Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia

**Departamento de Engenharia Electrotécnica**

# A Lightweight Distributed Super Peer Election Algorithm for Unstructured Dynamic P2P Systems

by

**Pedro Manuel dos Santos Henriques**

Dissertação apresentada na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa para obtenção do grau de Mestre em Engenharia Electrotécnica e de Computadores

**Orientador Científico**: Professor Adolfo Steiger Garção

**Lisboa**
**2011**

# Acknowledgements

First, I would like to express my deepest appreciation to all my friends who accompany me and all the support given during the course of this dissertation. The discussions, issues and the ideas proposed were always welcome and in many cases proved to be quite useful.

The biggest thanks goes to my family, for always giving me all the support I need, for making my life a lot easier when writing this dissertation, and for being always available, thanks to my mother Maria, my father Manuel and my sister Paula. The special thanks to my grandmother and grandfather who never stopped believing in me. And that always encouraged me to pursue what I desire and my dreams, with a huge smile in their faces.

Thanks to the support given by the GRIS for making this dissertation. For all the ideas discussed, the brainstorming, the problems outlined, the availability and the ease communication, thanks to Pedro Maló and Bruno Almeida.

Thank to my home colleagues (Pedro Lisboa and Gonçalo Pires), in addition to being great friends, are always present to give support in difficult situations and to help relaxing. Thanks to all the colleagues who collaborated in the review of the dissertation, and the availability of their free time to do so.

Thank you very much to everyone who contributed for making this dissertation . . .

, without you this would not be possible.

*Pedro Manuel dos Santos Henriques*

Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica
September of 2010

# Abstract

Nowadays with the current growth of information exchange, and the increasing mobility of devices, it becomes essential to use technology to monitor this development. For that *P2P* networks are used, the exchange of information between agencies is facilitated, these now being applied in mobile networks, including *MANET*s, where they have special features such as the fact that they are semi-centralized, where it takes peers more ability to make a greater role in the network. But those peer with more capacity, which are used in the optimization of various parameters of these systems, such as optimization\to research, are difficult to identify due to the fact that the network does not have a fixed topology, be constantly changing, (we like to go online and offline, to change position, etc.) and not to allow the exchange of large messages. To this end, this thesis proposes a distributed election algorithm of us greater capacity among several possible goals, enhance research in the network. This includes distinguishing characteristics, such as election without global knowledge network, minimal exchange of messages, distributed decision made without dependence on us and the possibility of influencing the election outcome as the special needs of the network.

# Resumo

Hoje em dia com o corrente crescimento de troca de informação, e com a elevada mobilidade dos dispositivos, torna-se essencial utilizar tecnologias que acompanhem este desenvolvimento. Para tal as redes *P2P* são utilizadas de forma a facilitar a troca de informação entre entidades, sendo estas trocas agora aplicadas a redes móveis, nomeadamente *MANETs*. Estas possuem com características especiais, como o facto de serem semi-centralizadas, onde se utilizam os nós com maior capacidade para terem um papel mais preponderante na rede. Mas estes nós com elevada capacidade, utilizados na optimização de vários parâmetros destes sistemas tais como as pesquisas, são de difícil identificação, devido ao facto de a rede não ter uma topologia fixa, e estar em constante mudança, (tais como nós possuírem o estado de online e offline, e a sua mudança de posição na topologia da rede, etc.) e não permitir trocas constantes de mensagens. Para tal, este trabalho propõe um algoritmo distribuído, de eleição de nós com maior capacidade, que entre várias possíveis finalidades, optimizam as pesquisas na rede. Este algoritmo conta com características diferenciadoras, como a eleição sem um conhecimento global da rede, a troca de um baixo número de mensagens, a decisão ser efectuada de forma distribuída sem dependência de nós e a possibilidade de influência do resultado da eleição consoante as necessidades especiais da rede.

# Nomenclature

| | |
|---|---|
| **3GPP** | **3rd Generation Partnership Project** |
| **5G** | **Fifth Generation** |
| **DHT** | **Distributed Hash Table** |
| **DML** | **Dynamic Layer Management** |
| **FP7** | **Seventh Framework Programme** |
| **GML** | **Graph Modelling Language** |
| **GNP** | **Global Network Positioning** |
| **ID** | **IDentifier** |
| **IEC** | **International Electrotechnical Commission** |
| **IEEE** | **Institute of Electrical and Electronics Engineers** |
| **IoT** | **Internet of Things** |
| **IP** | **Internet Protocol** |
| **IPTV** | **Internet Protocol television** |

| | |
|---|---|
| **ISO** | **International Organisation for Standardization** |
| **ISP** | **Internet Service Provider** |
| **JXTA** | **Juxtapose** |
| **LST** | **Lightweight Super Peer Topologies** |
| **MANET** | **Mobile ad hoc Network** |
| **NAT** | **Network Address Translation** |
| **OSI** | **Open System Interconnection** |
| **OWL** | **Web Ontology Language** |
| **P2P** | **Peer-to-Peer** |
| **PSO** | **Particle Swarm Optimization** |
| **SPC** | **Super Peer Candidate** |
| **TCP** | **Transmission Control Protocol** |
| **TTCN** | **Tree and Tabular Combined Notation** |
| **TTL** | **Time To Live** |
| **VCoP** | **Virtual Communities of Practice** |

| | |
|---|---|
| **VO** | **Virtual Organisation** |
| **VoIP** | **Voice over Internet Protocol** |
| **VSM** | **Vector Space Model** |
| **W3C** | **World Wide Web Consortium** |
| **WiMAX** | **Worldwide Interoperability for Microwave Access** |
| **XML** | **Extensible Markup Language** |

# Contents

# Index of Figures

# Index of Tables

# CHAPTER 1 Introduction

In the past 50 years, our world has experienced the most important changes and evolutions in centuries. In different scenarios such as social, economic and political development has brought the world to new set of paradigms of living, education and social interaction. The expansion of the Internet into previously inaccessible locations, led to the form of a worldwide network of interconnected computer networks based on the *TCP/IP* standard communication protocol. This network is becoming a fundamental part of the daily-life of our society, and was driven over last 30 years by the exchange of data between server platforms, personal computers, mobile devices and sensors was described in (Alcatel-Lucent, 2009).

The internet has become a global critical infrastructure and a remarkable catalyser to the used for innovation and growth. It is the "nervous system" of our society and one of the most critical infrastructures of the 21st century. The internet fast evolution is described in (Georgios Tselentis, 2009), where it is described that at "*Fifteen years ago nobody would have envisaged the Internet as it is today as well as its various applications. Some remarkable cases can be outlined such as i) the Web, which processes 100 billion clicks per day and offers 55 trillion links between hypertext pages, ii) the exchange of 2 million of emails per second and iii) instant messengers with 1 million messages send per second.*" It is massively used on supporting social and economic development and has enabled multiple waves of innovation: first with the introduction of the web, then with the integration of communication and audio visual services (*VoIP* and *IPTV*) and recently with the establishment of multiple online services and applications.

If the current growth continues to be registered, the internet will certainly overtake TV as the most consumed form of media for the first time, since it has already surpassed all other media. Internet is already the most important information exchange mean and has become the core communication environment not only for business relations, but also for social and human interaction. Many of the key trends today are focused on Social Networking (*LinkedIn, MySpace, FaceBook, Twitter*), 3D Real Time (*Sims Online, Second Life, ActiveWorlds*), Electronic Media (*E-Media*), Internet of Things (*IoT*) (The Internet of Things, November 2005) and entertainment distribution Systems, amongst others (Alberti, 2009).

Social networking will replace e-mail as the primary means of communication for some business users in some stage over the future, since everyday life the email is becoming more integrated with the massive growth of mediums. A new trend is emerging with social networks created to help its members: Virtual Communities of Practice (*VCoP*).

*VCoP*s build on existing formal content tools such as portals, learning management systems, document management, content management, and knowledge management, as well as team and productivity tools. They enable organisations to add a new dimension-the informal organisation. VCoPs are distributed groups of people who share a common concern, problem, mandate, or sense of purpose. "*They can be used to facilitate the informal knowledge transfer that drives leadership development, productivity, and innovation*", as can be described in (Sauve, 2007). Accessing social networking sites on mobile is also an increasing trend worldwide (Associates, 2009). Beside the ability to browse the internet, mobile phones also know where they are, thanks to in-built GPS satellite technology or triangulation from mobile phone masts. They now can tell if others phones are in the same area, and inform if someone is near you, in order to access a lot of information about them, providing the perfect ingredients for real social interaction (Harvey, 2008).

The future in 3D internet is going to look a lot more like an interactive movie than a newspaper in a little while, and 'virtual worlds' are just convenient shorthand for the first step. As processing power and bandwidth increase, it will become possible to represent things in interactive 3D much more cheaply and easily once technology allows mainstream deployment. With the realistic 3D fully-textured and animated avatars it's now possible to develop social (and even clinical) skills and socialise and interact with other people (Chevalier, 2005). Moreover, with the use of 3D Real Time, the level of interactivity between users will increase and with the use of entertainment distribution systems, the distribution of radio and television programmes, movies, music, ring tones, games, and various data applications to the general public is today possible via a variety of dedicated networks and special end user terminals.

In this evolving environment, user-generated/user-centric rich content as well as community networks and the use of *P2P* overlay systems are expected to generate new schemes of interaction and cooperation and be able to support new innovative applications, like virtual collaboration environments, personalised services/media, and virtual sport groups.

In the described context, the interaction with content combined with interactive/multimedia search capabilities across the distributed repositories and *P2P* (also mobile) infrastructures and the dynamic adaptation to characteristics of diverse terminals are expected to contribute towards content creation and electronic media delivery (Zahariad, 2008). For the near future is then expected that people use the internet to produce, and seamlessly deliver and share their own multimedia content. In a few years everyone will be a multimedia content producer (by publishing digital pictures, video recordings, remote e-health services, home surveillance, etc.), multimedia content mediator (by storing/forwarding streaming content) and multimedia content consumer (digital television, video on demand and mobile broadcasting).

Regarding this mobile and the current ubiquitous environments the data may be transferred and disseminated among several system entities. Such data can be generated by sensors, refined and reasoned by processors, and consumed by clients, using highly distributed system that uses *peer-to-peer* computing (Ekaterina Chtcherbina, 2002). Little gadgets (*PDAs*, *Smart Phones*, *Tablets*) are now always connected to the network, and have become an integral and intimate part of everyday life for many millions of people, extending the actual internet, because of the structure created by these heterogeneous devices called the *"Internet of Things"* (*IoT*) (Gérald Santucci, 05 September, 2008). The *MP3/MP4 Player*, which started as just a music player, has now wireless communication capability. Mobile phones also already have more memory, faster processors, larger screens, and higher network bandwidth than ever before.

The current mobile networking associated with the *P2P* computing system enables the user to synchronize contacts, calendar and *Gmail* account, for real-time mobile information. This system also allows using features as *peer-to-peer* communications and proximity-based social interactions (Alliance, 2008). In addition to the aforementioned devices, smart books will have the possibility to interact with the reader when this is reading in real time, providing in the page more information according to the topic. Many of the content requested by the users will be available online such has live video. At the same time, numerous new wireless network technologies are being developed.

Standardization organisations like 3rd Generation Partnership Project (*3GPP*) and *WiMAX*[1] Forum are already in advanced stages of standardizing new mobile wireless networks and several industry players are taking part in long-term research with the focus of developing future mobile infrastructures. While so-called fourth generation (*4G*) mobile networks are often seen as focusing on seamless integration of multiple wireless technologies, the majority of actual new mobile networks currently being developed by the industry are focusing on improving packet data access; increasing throughput and capacity while reducing end-to-end delays and delivery costs (December 2003).

The next mobile technology called 5G Technology is going to be a new mobile revolution in mobile market. The 5G technology has extraordinary data capabilities and has ability to tie together call volumes and data broadcast within the latest mobile operating system. 5G Technologies have an extraordinary capability to support Software and Consultancy. The 5G technology distributes internet access to nodes within the building and can be deployed with union of wired or wireless network connections. 5G would be about "ubiquitous computing", that is, having the ability to access the applications the user wants from any platform, anywhere, any time. The current trend of 5G technology has a glowing future (Mehta, 2009).

Live video streaming services have spread quickly over the Internet. Video sharing websites such as *YouTube* attract millions of users per day, and a large number of TV channels are already available on the Internet or through *IPTV* services provided by *ISPs*. *Peer-to-peer* live video streaming, or *P2PTV*[2], greatly reduces the bandwidth requirements of the source by making users serve part of the stream to other peers downloading the same content. *Peer-to-peer* live streaming systems allow a bandwidth constrained source to broadcast a video feed to a large number of users. In addition, a design with high link utilisation can achieve high stream rates, supporting high quality video. The IT industry now deals which hundreds of millions of objects that are connected to the networks.

In addition to accessing the Internet, the sheer amount of such devices equipped with short-range radio interfaces (such as *Bluetooth*) and with the higher mobility and self-configuring capacity could be organised to form *MANETs* (Fernanda P. Franciscani, 2004).

[1] Worldwide Interoperability for Microwave Access

[2] Refers to peer-to-peer software applications designed to redistribute video streams in real time on a P2P network.

A *MANET* is a self-configuring network composed of mobile routers (and associated hosts) connected together by wireless links—the union of which form an arbitrary topology. Nowadays peers can be already placed everywhere, even when the infrastructure is weak or absent, and even if the peers are mobile. One current example is to use the *MANETs* to provide communication to the mobile phones even if their users are in the subway system in any major city. The attractive solution is to use *MANETs* where communications can reach phones that are outside the direct reach of the cell towers, because other nearby phones can act as intermediaries. In this way, phones that are still within cell tower coverage pass on relevant data, via a series of local hops, to phones that are further and further away from cell tower coverage. Moreover, it saves power as communication is performed with nearby devices and not with a distant cell tower.

Over the last years the interesting features of *P2P* networks such as self-organisation, scalability and robustness have captivated the interest of researchers. *Schollmeier* (Schollmeier, 2001) defines a *P2P* infrastructure as a distributed architecture where participants share a part of their resources (processing power, storage capacity, and printers) and where these resources are also accessible by other peers directly, without passing through intermediary entities. Therefore, participants of *P2P* networks are providers and consumers (clients and servers) of resources simultaneously. This architecture model is different than a client/server model where each entity acts as a client or as a server, but never as both at the same time. All computer systems can be classified into centralized and distributed. Distributed systems can be further classified into the client-server model and the *P2P* model. In a hierarchal model, the servers of one level are acting as clients to higher level servers.

Examples of a flat client-server model include traditional middleware solutions, such as object request brokers and distributed objects. Examples of a hierarchical client-server model include DNS server and mounted file systems. The *P2P* model (Dejan S. Milojicic, 2003) is described as being either *pure* or *hybrid*. In a pure model, there does not exist a centralised server. Examples of a pure *P2P* model include *Gnutella* (Q. Lv, 2002). In a hybrid model, a server is approached first to obtain meta-information, such as the identity of the peer on which some information is stored, or to verify security credentials. From them on, the *P2P* communication is performed. There are also intermediate solutions with high-order peers, such as *KaZaA* (J. Liang, April 2005). High-order peers contain some of the information that others may not have (J. Liang, April 2005) (Dejan S. Milojicic, 2003).

Within the last years, *P2P* technology has seen an explosion of new devices that take the rule of peer, each with unique features. It is impossible to classify these peers generally as they are very different. In the past these available peers on the network were only restricted to computers, and others few devices for exchange information.

So our future networks need to be extremely cost-effective to run, capable of performing under the pressure of tremendous traffic growth, and doing all this without degrading or over-complicating the user experience. *P2P* are inherently scalable and reliable because of the lack of a single point of failure. *P2P* systems are robust against global, catastrophic failure, although single nodes may fail. The high mobility in *MANETs* gives them the features to be the ideal choice for the challenges that society imposes to the future network.

## 1.1 Scope

Since peers depend on each other, for getting information, computing resources, forwarding requests. which are essential for the functioning of the system as a whole and for the benefit of all peers, researchers and developers have dedicated themselves optimise and improve of the *P2P* structure. This occurs much because of the increasing need to establish communications between huge numbers (millions) of mobile/fixed devices, with highly heterogeneity.

In addition, the *P2P* systems may in some cases be affected by lack of efficiency and poor performance if they do not address their peers' heterogeneity but also if they do not apply an approach to adapt their structure to the properties of individual peers (Ling-Jyh Chen, 2006).

There are approaches to exploiting heterogeneity in *P2P* systems, in which the system structure is in line to the properties of the participating peers. Such approaches include spanning tree and mesh structure optimisations in streaming systems, adaptive *peer-to-peer* topologies (Tyson E. Condie, 2004) and message flow adaptation in *Gnutella*, and the introduction of virtual servers to distributed hash tables. Also super peer networks maintain a balance between the inherent efficiency of a centralized search mechanism, and the autonomy the load balancing and the robustness to attacks provided by distributed search. Furthermore, super peers networks take advantage of the heterogeneity capabilities (e.g., bandwidth, processing power, etc.) across the existing peers, which recent studies have shown to be enormous.

Hence, new and old *P2P* systems like *KaZaA* (J. Liang, April 2005) (Beverly Yang Hector, 2002) and *Gnutella* are adopting super peers in their design for handling search, has also well-known telephony systems like *Skype* (S. Guha, 2006) (Schulzrinne, April 2006) and *P2P-SIP* (Erkki Harjula, 2006).

A super peer is a peer in a *peer-to-peer* network that operates both as a server to a set of clients, and as an equal in a network of super peers. The super peers are peers that have agreed to cache advertisement indices (i.e. pointers to edge peers that cache the corresponding advertisement). Super peers conceptually correspond to well-known locations used for the principal purpose of indexing and locating advertisements. In a super peer network, more powerful machines with faster connections are used has super peers. These super peers receive (or generate) queries, try to answer them, and then forward them to all connected peers. The edge peer tries to answer queries but never forward them to others peers.

Super peer network supports sophisticated routing and distribution strategies, and are desired to be used in preparing the ground for advanced mediation and clustering functionalities of stable peers that are less subject to churn. The structure is governed by certain rules, and consists of elements with a predefined set of features and functions has described in "*Designing a Super-Peer Network*" (Beverly Yang Hector, 2002). The super peers are used to build up the routing backbone for the whole network, as they are used to index information and improve the search, and the edge peers for inquiry the network. For example, *KaZaA* uses super peers to index data stored by clients and to handle the search protocol.

Also studies of the *Gnutella* and *Napster* systems, emphasize that the user population of both systems included significant heterogeneity in the available bandwidth of the participants and concluded that this heterogeneity should be exploited by selecting peers with greater capabilities and a high level of responsibility.

The super peer systems have higher search efficiency because, instead of all the peers in the system, only super peers have active participation in the search processes. This leads that, an appropriate layer size ratio is required, that is the ratio of the number of leaf peers to the number of super peers.

## 1.2 Problem

Within *MANETs*, super peers are used for the purpose of data aggregation under the conditions that are well distributed among the peers in order to set a global ratio of super peer to non-super peer. Super peers function as index servers for the edge nodes, and are used to propagate the queries to their final destination. They are also maintained to meet application-specific performance requirements, and deal with the remaining battery life of the connected devices (peers). It can occur that when the battery runs out on some devices, and consequently the network topology undergoes significant changes that should be taken into consideration in maintaining the super peer network and ratio.

Having a super peer network applied to *MANETs*, the network has also to taken into account the *ad-hoc* properties of peers, and their highly mobility. Having a network that is self-configuring and that the peers are free to move independently in any direction, they will therefore change the connection links between them. The mobility affects the super peer network since those mobile peers that are present in a particular “position” when the super peer topology is set, may not be in the same “position” after a short amount of time, and the previous established super peer topology might not be valid anymore. The peers moving inside the network is a challenge to the establishment of super and edge peers, since the super network needs to adapt it structure to the present topology in the network in order to respond according.

Despite this mobility and *ad-hoc* properties another restriction on large scale *MANETs*, is over the number of messages exchanged. A small number will avoid flooding the network or inducing mechanism that may reach a deadlock, and need to launch additional phases to deal with this situation, and the possibility of lost or even the arriving of invalid messages. In all peer any running process needs to be lightweight CPU to allow the process to run over any peer without high processing power or memory.

Also on large scale *MANETs* the global view over all the present peers is impossible to reach, so creating and maintaining the super peer structure, leads to the use of distributed computing processes, to achieve the common goal.

The possibility is to establish the adequate super peer topology over a *MANET*, and to perform an adaptive mechanism, that aims to promote the super peers in accordance with the topology presented by the network. **This work proposes to study the super peer appointment problem and to propose a solution in order to keeping the super peers topology over a *MANET*.**

## 1.3 Methodology

The methodology adopted in this work (

Figure 1-1) is based on the basic principles of scientific method, which involves the following general steps (Schafersman, 1994):

1. Defining the issue/problem;
2. Collection of information and resources;
3. Forming a hypothesis;
4. Preparation of an experiment;
5. Analysis of results and conclusions;
6. Publication of results.

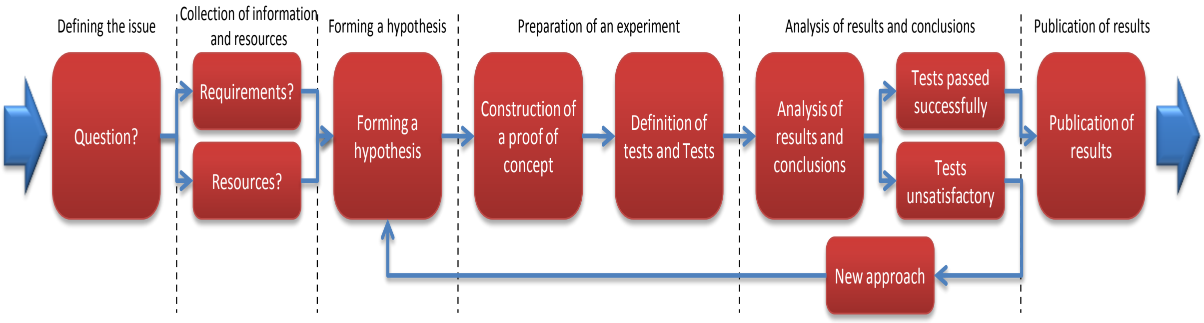


Figure 1-1 - Research methodology

### 1. Definition of issue/Problem

The work begins with the definition of a problem and with the introduction on the *MANETs* and the *P2P* infrastructure. The question of this dissertation comes from developing a Lightweight Distributed solution for enabling the realization of the Election mechanism of Super Peer over a *MANETs* in a decentralized manner.

### 2. Collection of information and resources

It was necessary to raise (*P2P*, *MANETs*) the key concepts, and analyse what exists today in the literature and that fits the problem raised. It was also necessary to examine the different approaches that could be taken to resolve the problem and to a better understanding of it. Because the issue is framed in mobile P2P networks, it is necessary to examine these technologies and to examine some limitations.

### 3. Formation of a hypothesis

Based on the requirements defined and the available resources, it follows for a conceptual achievement that serves as the research hypothesis. This is the basis for the solution, and should enable both a conceptual approach, namely an analysis of the problem of an abstract point of view, but also a specification in order to be able to leave for the elaboration of an experience.

### 4. Development of an experiment

The hypothesis is subject to a validation process through experimentation. For such an implementation it will take place on the form of a proof of concept, in order to perform tests to validate the hypothesis proposed in answer to the question set.

### 5. Analysis of results and conclusions

To assess the proposed solution, it´s performed a set of tests to check whether it complies with the requirements set. These tests are applied in a controlled environment through testing to validate the solution. If the tests fail, back to point 3, and proceeds to the refining of the solution, until there is a new solution that successfully meets the defined requirements. In the final conclusions are settle, based on results obtained from the analysis.

### 6. Publication of results

Upon the verification of results, publications shall be made to expose the results. For more detailed description over the publications, the Chapter 6 should be consulted.

## 1.4 Outline

The dissertation is comprised of six chapters, the first being the current introductory chapter. The remaining is organized as follows.

Chapter 2 evaluates background information and related work in *MANET*s (Mobile Ad Hoc Networks), their utility and use, over *P2P* infrastructure with particular emphasis to super peer networks. Over super peers networks the super peers election and the super peer eligibility is described. This chapter only serves to provide general information over basics concepts, needed for a clear and concise understanding of the subject discussed in this dissertation.

Chapter 3 introduces the study over the existing super peers' criteria and election algorithms. In this chapter it´s also highlighting the achievements and a comparison (advantages and weaknesses) of existing algorithms found in the literature review, regarding the Super Peers Election Algorithms. Thus, a framework is performed on the super peers' election techniques, as the limits of the main theme of this dissertation.

Chapter 4 formally presents the details regarding the implementation and decisions made for the prototype. The core functions, the solutions taken, and the algorithm steps. The specification of the algorithm is also discussed here.

Chapter 5 describes the adopted methodology and the proof of concept, the tests definition and the implemented solution, along with the performance evaluations and the analysis over test executions and obtained results.

Chapter 6 verifies the achievement of the initially objectives set forth this dissertation, through the analysis of the test results, and discusses potential direction for future research, regarding the obtained results.

# CHAPTER 2 Background

This dissertation focuses on the area of Super Peer Election Problem, over a *MANET*. The purpose of this chapter is to introduce the main concepts and definitions to the inherent aim of this dissertation. This Background Chapter is intended to provide the reader with an initial set of concepts, and all the definitions in order to provide a broader view over the area where this dissertation is focuses.

## 2.1 Mobile Ad Hoc Networks

A *MANET* enables wireless communications between participating mobile nodes without the assistance of any base station. In a *MANET*, each mobile terminal is an autonomous node, which may function as both a host and a router. In other words, besides the basic processing ability as a host, the mobile nodes can also perform the switching functions as a router. So usually endpoints and switches are indistinguishable in a *MANET*. For example, in Figure 2-1 there is no direct radio channel, between the PDA and the Webcam. In this case is used terminals with relaying capabilities (such as the laptop the cellular phone, and the headphone), that could serve as intermediate routers and support the connection between the two devices.

Figure 2-1 - A Mobile ad hoc network

This network is an autonomous system of mobile routers (and associated hosts) connected by wireless links (the union of which form an arbitrary graph). A *MANET* may operate in a standalone fashion, or may be connected to others networks.

This type of network has several salient features, considering five very evident: *Dynamic Topologies*, *Bandwidth-Constrained, Variable Capacity Links, Energy-Constrained Operation and Limited Physical Security*. For Dynamic Topologies the nodes "*are free to move randomly and organize themselves arbitrarily thus, the networks wireless topology may change rapidly and unpredictably"* (Sun, 2001), and thus the network topology, and may consist of both bidirectional and unidirectional links. *Bandwidth-constrained, variable capacity links* are burdens applied to this type of network derived by the very nature of the network, where wireless links have significantly lower capacity than their hardwired counterparts, and also the realized throughput of wireless communication, is often much less than a radio´s maximum transmission rate. In Energy-constrained operation, all or some of the nodes present in *MANET* may rely on batteries or other exhaustible means to use has energy source, at this limits some of the system design criteria, that will need to be optimize by the energy conservation issue. Finally, the *Limited physical security* relates to the fact that mobile wireless networks are generally more prone to physical security threats than are fixed-cable nets. However, in opposition the decentralized nature of the network control in *MANET´s* provides them the additional robustness against the single points of failure of more centralized approaches. (S. Corson, 1999).

Wireless ability is a very important part of communication technology that supports truly pervasive computing, because over different contexts, information exchanged between mobile units cannot rely only on a fixed network infrastructure, but on rapid configuration of wireless connections taken in real time. When two nodes are out of one another's transmission range, it´s required additional intermediate nodes to establish the communication, which use relaying messages to set up the communication between each node. The broadcast operation is the most fundamental role in *MANETs* because of the broadcasting nature of radio transmission: "*when a sender transmits a packet, all nodes within the senders transmission range will be affected by this transmission, so if one node transmits a packet, all its neighbours can receive this message*" (Wu, 2007).

The decentralized network uses control messages that are used to disseminate the necessary information for network organisation (Subbarao, 1999) the routing execution is accomplish by the nodes themselves. There is no background network for the central control over the existing network operations, and since the network relies on multi-hop communication, the control and management over the network is distributed among the nodes. "*These networks have a distributed communication architecture, where nodes make individual decisions on routing and medium access*" stated in (Marcel C. Castro, 2009).

## 2.2 Peer to Peer Systems

*Peer-to-Peer* systems have emerged as the favourite option to share huge volumes of data. In such settings, one objective is the efficient search across peer databases by processing each incoming query without overly consuming bandwidth. One of the most fundamental properties over nowadays existing large-scale *peer-to-peer* systems is a very high heterogeneity and dynamism of the peers participating in the system. "*A pure P2P network does not have the notion of clients or servers but only equal peer nodes that simultaneously function as both "clients" and "servers" to the other nodes on the network*" was described in (Innovations, 2007). The network can be extremely large, scaling to millions of peers, with levels of network membership that can be highly unstable. The connected peers construct a virtual overlay network on top of the underlying physical infrastructure, where no peer has a global view of the system, and a global behaviour emerges from local interactions. The use of *peer-to-peer* applications to content sharing makes use of each individual offering in order to provide large-scale distributed services in the network, since applications designed for these networks must distribute fairly their loads among all the nodes and optimize the consumption of limited resources.

Depending how the *peer-to-peer* network topology is organized, the *peer-to-peer* architectures can be divided into structured and unstructured architectures. Unstructured architectures can be further divided into centralized, decentralized, and semi-centralized architectures. The main difference between unstructured and structured architectures is that in structured architectures peers form a defined structure, or a topology, that has to be kept up as nodes join and leave the network.

In unstructured peer-to-peer networks, the network presents itself in a more a construction-free (Resource Location in P2P Systems, 2009). "*Unstructured overlays do not impose a rigid relation between the overlay topology and the indices/resources placement, since flooding or random walks are used to locate resources*" as described by (Marcel C. Castro, 2009).

Semi-centralized architecture is a combination of the centralized and decentralized *peer-to-peer* architectures, thus it is often called hybrid architecture. In the semi-centralized architecture, there are two kinds of nodes: edge nodes and super nodes. The super nodes are connected to each other in a similar fashion to nodes in the decentralized *peer-to-peer* architecture. The edge nodes are connected to the super nodes in the centralized *peer-to-peer* fashion.

### 2.2.1 Semi-Centralized Architecture

The semi-centralized architecture combines good features from the two *P2P* models, the centralized and the decentralized in order to achieve a hybrid model. The peers in the super peer network were divided into two layers: the Super-Layer and the Edge-Layer. Each peer in the Super-Layer, is called super peer, and is responsible for propagating the queries on behalf of the others peers in the Edge-Layer called edges, for search of information and routing of data (Anis Ismail, et al., 2009) (Beverly Yang Hector, 2002). Figure 2-2 illustrates the topology of a super peer network.

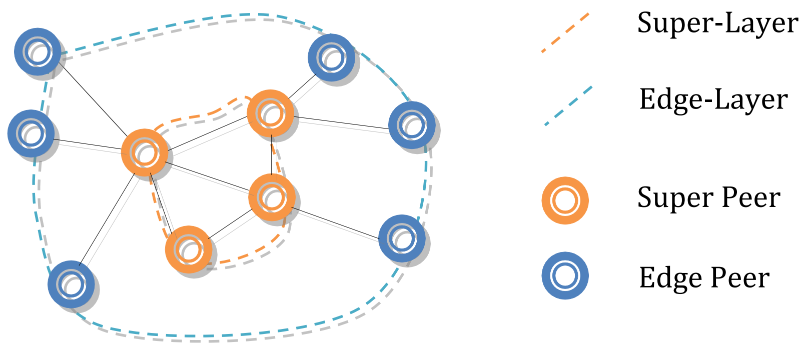


Figure 2-2 - Two-Layer Network, with the Super-Layer and the Edge-Layer

In the semi-centralized architecture, the super nodes function as index servers for the edge nodes. When an edge node joins the network, it connects to a super node and uploads the list of its resources to the super node. When an edge node searches for a resource, it first sends a query to its super node. The super node will then transmit the query further to other super nodes using a similar algorithm that is used in decentralized *peer-to-peer* networks.

The query does not have to be flooded to edge nodes because the network of super nodes, as the index servers, provide a distributed cache approach, that represents the total knowledge of the resources available in their edge nodes. After the reply comes back to the querying edge node, it connects directly to the other edge node hosting the queried resource (Ehrich, 2009). The use of super peers facilitates the searches due to their capacity to keep the information in cache. Caching is done in super peers through files containing metadata about information stored in each node. The search is then performed without the need to flood the network, but being referred only to the super peers, causing the message traffic to be lower and the query is processed more quickly (Fonseca, 2008).

With the use of super peers for data caching nodes, they are responsible to retrieve data from the data source or other data caching nodes, and then can directly provide the data to the requested nodes. The super peers also cache pointers to files recently requested by their edge peers. The use of semi-centralized architecture approach intend to reduce the traffic and to achieve better scalability by providing a hierarchical structure since super peers act as a proxy to the clients connected to it. It has been observed that the characteristics of the peers such as storage, bandwidth, etc., vary by several orders of magnitude between single peers, while a large fraction of peers have relatively few resources small subsets of peers possess significant fractions of the total system resources.

Hence, super peer networks introduce two basic actions in addition to query: *joins* (for which there is an associated *leave*), and *updates*. When a client wishes to submit a query to the network, it will send the query to its super peer only. The super peer will then submit the query to its neighbours as if it were its own query, and forward any "Response message" it receives back to the client. Outside of the cluster, a client's query is indistinguishable from a super peers query (Beverly Yang Hector, 2002).

Further capabilities like mediation and transformation of queries and answers also can be implemented in a super peer. Connections between super peers serve to reduce the network diameter and make these services more efficient. The super peer in different context of the literature is defined as *super-peer*, *supernode*, *super-node*, *ultrapeer father nodes* and as *Super Peer*.

Because well-designed super peer networks promise large performance improvements for *P2P* systems, it is important to base election decisions on a solid understanding of the systems behaviour. Super peers allow decentralized networks to run more efficiently by exploiting heterogeneity and distributing load to machines that can handle the load.

On the other hand, the semi-centralized architecture is not totally free of the flaws of the client-server model, as it allows multiple separated points of failure increasing the health of the *P2P* network. "*The super peer paradigm is not limited to file sharing: it can be seen as a general approach for P2P networking. Yet, the structural details are strongly application-dependent so it's impossible to identify a "standard" super peer topology*" in (G. P. Jesi, 2006).

## 2.3 Election in P2P Systems

The election means "to choose or make a decision". The election involves a population, where by applying the act of voting, elects a special element amongst them, to serve a special role of “master” or “leader”.

An election is a decision making process that involves the selection of a subset of the peers (population) to be elected and to serve a special role. The peer might be well-dispersed throughout the *peer-to-peer* overlay network, or may be concentrated in a particular zone, but in any case they have to fulfil present requirements to be able to participate in the election.

The election problem is highly challenging because in the *peer-to-peer* environment, a large number of peers must be elected from a huge and dynamically changing network in which neither the peers characteristics nor the network topology are known a priori. In an election all nodes in the network, compete to perform a particular function. Since anyone can participate, all of them can be part of the election process.

### 2.3.1 Super Peer Eligibility

A node might not be eligible to be selected unless it meets certain minimum qualifications. Those qualifications vary according to the peer utility in the system, and according to the election propose. The election comes from the need for a peer with special characteristics that must be elected among several others on the network, in order to address specific needs of the network. To address the peers’ heterogeneity, many selection mechanism use **Thresholds** to distinct between a candidate to super peer, in opposition to the others peers with few resources and that will only set the role of edge peers. To use the available parameters each peer aggregates all information related to these parameters, on a common utility metric named “$U$” (Jim Dowling, 2006).

With the use of this utility metric, the super peer selection problem can be solved by calculating a super peer utility threshold. The super peer set selected this way is optimal in the sense that the utility of peers in the set is maximised. Each super peer has a higher utility than any edge. The number of selected super peers is directly controlled by the super peer utility threshold. A number of different criteria can be applied when calculating super peer thresholds. For some simplest cases, the threshold can be explicitly given by a higher-level application.

In the **Super Peer Utility** the utility "$U(p)$" of peer $p$ is a number that reflects the appropriateness of peer p to act as a super peer (Nao Chen, 2010). The higher the peer utility, the more suitable a peer is to occupy the super peer position. In such applications, peer utility can be defined as a function, such as a weighted sum or product, of hardware parameters. With this approach the computation of the peer utility is simple and straight-forward. Each peer can obtain from the operating system, or measure directly, the values of its relevant parameters and independently compute its utility.

Similarly, a **Proportional Threshold** (Bartosz Biskupski, 2009) is defined as a utility value, $t_Q$, such that a fixed fraction $Q$ of peers in the system has utility greater than or equal to $t_Q$. In a system with $N$ peers, a proportional threshold is described by the following equation

$$D\,(t_Q) = Q \cdot N \tag{2.1}$$

This proportional threshold allows the peers to adapt the number of super peers present regarding the total system size. As the system grows and shrinks in size, the proportional threshold increases and decreases, in order to adjusting the number of super peers in the system so that the ratio of super peers to ordinary peers remains constant (Bartosz Biskupski, 2009). For many others proceedings, the desired number of super peers depend not only on the system size but also on the capabilities of available peers (Yifen WEI, 2007). Systems such as *SG-1*, *SG-2* and *DLM* introduce the a new concept referred to peer capacity and generate super peer groups that have sufficient capacity to support all other all remaining peers as edge.

Two common criteria used by many selections mechanisms for selecting super peers are the **Capability and Stability,** since these two variables can also be the two variables used to classify the peer for the heterogeneity (Jim Dowling, 2006). For many applications, peer stability is amongst the most important peer characteristics, since in typical *P2P* systems the session times vary by orders of magnitude between peers, and only a relatively small fraction of peers stay on-line for a long time.

Stability was already been described in the literature by (Yuh-Jzer Jounga, 2009) stating that: "*Stability measures how stable a peer is. Stable peers are unlikely to fail, or to join/leave the system very frequently, they often have long session time. Capability is a comprehensive evaluation of hardware and bandwidth of a peer. It can be measured by simply mapping hardware specification and/or bandwidth to some predetermined scores*".

The peer capacity, usually described by "$C(p)$", refers to the total amount of resources, (storage space, processing power, and bandwidth) accessible at the peer, or the maximum load the peer $p$ can handle at a time, while the peer load, represented by "$L(p)$", represents the total amount of resources that are currently in use. The selected approach involves defining the peer utility as a function of the total peer capacity, and to collect information about the system load to perform the threshold over the super peer selection. This approach allows to the peer utility, and subsequently the system topology to remain stable, while the current super peer´s may be increased or decreased as the total system load accompanies the growth and the decreasing, as the nodes join and leave the system. In (Bartosz Biskupski, 2009), the client threshold as defined by the below equation. In witch $D(t)$ is the number of selected super peers, $N - D(t)$ is the number of clients, and $D^c(t)$ is the total super peer capacity.

$$D^c(t) = N - D(t) \tag{2.2}$$

In the case that the peer utility is defined as peer capacity, this leads to $U(p) = C(p)$ for every peer *p*, then the super peer set generated by the use of the clients threshold is equivalent to the one in the *SG-1* target topology, has evidenced in (Montresor, 2004). In the opposed case where $U(p) \neq C(p)$, the existing super peers are selected from the highest-utility peer currently in the system and the select peer´s to take the role of super peers is determined by the two variables present in the system: the system size and the super peer capacity.

Other more general approaches, collected from the current literature derivate from the use of selecting the super peers using the concept of peer load that led to the emergence of the **Load Threshold**. This concept can be used in accordance with the objectives desired by an application, but generally it can represent connected clients, stored data, network transfers, handled requests, running jobs, or other application-specific concepts (Heng Tao Shen, 2004). The load threshold is defined as a utility value "*t*", that established that peers with the utility value above the t value, have a total capacity equal to the total system load. This is represented by the following formula (Changyong Niu, 2005):

$$D^C(t) = \sum_p L(p) \tag{2.3}$$

For any desired system to demonstrate efficiency, it's critical to maintaining an appropriate **Super Peer Ratio**. The *Super Ratio Maintenance* (SRM) is based on scaled comparison to defined dynamic promotion and demotion policies, and on *Particle Swarm Optimization* (PSO) to adjust the policies in the face of fluctuations in the capacity distribution.

This type of selection it is view as a decentralized, fault-tolerant and accurate solution in (Marc Sánchez-Artigas, et al., 2008). The previous approaches are used in order to distinguish between the many heterogeneous peers present in a large-scale network.

## 2.4 Summary

In this Chapter first was defined the *MANET* and the *P2P* approach. Then, over the existing *P2P* approach this architecture is analysed and decomposed in its two major components: the structured and unstructured architectures. The unstructured *P2P* architectures can be further divided into centralized, decentralized, and semi-centralized architectures. Next was shown how they are defined, their structure and organisation.

Semi-centralized architectures selects the super peer amongst all the peers present in the network to be used for indexing the queries, provided by the client (edge) peers. The election mechanism that aims to elect the super peer in the network is described, and next the peer differentiation is also described. Subsequently, was described several approaches to differentiate the peers. All the existing approaches for the super peer election will be described below in Chapter 3 to make a precise state of the art in the super peer election methods.

# CHAPTER 3 Study of Super Peer Election Methods

This chapter defines the challenges that the election mechanism has to take into account. The challenges are driven from the nature of the network, and related to the environment (*P2P* systems over a *MANET* structure) where the algorithm is intended to be applied. There will be also discussed the current implementation of the super peer election problem on regarding the different types of existing systems, their purposes, their main characteristics, and also their focus points. There will be a survey on the existing election mechanisms and by the final of this chapter is discussed the relation between the challenges and the existing super peer election algorithms.

## 3.1 Super Peer Election in MANETs Criteria

As a starting point the algorithm is intended to run in *MANET*s. In addition to the features and restrictions associated with *MANET*s, it will also be taken into account that the algorithm was developed to be used in *peer-to-peer* systems that have a semi-centralised architecture (through the use of super peers).

Some of the challenges are driven from the restrictions imposed by the network characteristics where the algorithm will be executed. In *MANETs context*, and due to the mobility of the nodes, additional messages for control and routing are also needed to establish the connection between the nodes. For the *MANET* it's necessary to use extra messages to control de network topology. For dealing with this the proposed solution has to **minimize the use of exchanged messages** between the nodes. Also the use of too many control messages will condition and even possibly degrade the efficiency of the algorithm. The minimum number of messages exchanged, in addition to avoid making flooding the network also saves the battery on the devices. In large scale networks there is no global knowledge and it's unrealistic for a node to maintain information about all of the nodes present in the network. The execution of the algorithm besides dealing with the wireless feature of the network, and the mobility of the nodes, it also has to be designed to use only the nodes local knowledge.

With the use of local knowledge it's possible to employ the algorithm in large scale networks, without having the global view of all nodes present in the network. At the same time, and since only local information is used, the election can take place independently of the network size.

On the other hand, having a local knowledge of all the peers in the network is a scalable solution. The use of only **local knowledge** provides the mechanism with no central repository that induces a single point of failure in the election.

Without the use of a central control (decision) system, all the peers should participate actively in the election process, which represents a challenge in large-scale networks. This challenge of the super peer election for *peer-to-peer* systems lies in its **distributed decision making ability**. This ability fulfils the need of a fully distributed algorithm that operates under dynamic conditions (ad hoc and mobile network), and that also attempt to exploit the heterogeneous capacities (e.g., bandwidth, processing power, etc.) of the participating peers to improve performance and reliability for the entire network (Q. Lv, 2002) (S. Saroiu, July 2003). Using the heterogeneous peer capacities, associated with the distributed decision making ability, it's possible to elect the appropriate peer to be promoted super peer based on those two concepts. In a distributed decision making each node runs a process by which it makes relevant local observations and, on the basis of these observations, makes a judgment on which reconfiguration action is to be taken. In an ideal situation each nodes local observations would be entirely representative of a more global scenario, and the nodes would easily reach a final decision based on those observations.

Mobile *ad-hoc* networks by their nature are highly adaptive and their capacities are used for much different intents. The election mechanism needs to provide the possibility to deal with the purpose for which the network is being created or used. This implies that election process needs to be conditioned from the exterior, e.g. based on context information or application goal. This is made possible through the use of a module called **external conditional decision**. This module allows changing the election mechanism, without being necessary to induce changes in the mechanism itself. With this external conditional decision the election makes use of external input parameters that are in context with the network purpose, in order to allow those parameters to influence the election. This ability allows the mechanism to adapt/modify the output of the election mechanism according to different proposes.

If in a specific network the main utility to the super peers are to be elected according to a pre-defined feature, only a simple additional layer is required at the election. This gives the election the possibility to use *ad-hoc* layers according to the specific network propose given to the use the super peers. This will provide the election mechanism with a more flexible nature, regarding the election network purpose.

## 3.2 Super Peer Election Methods

A general overview to relevant the super peers' election methods are provided in this section. There is a brief description of each algorithm regarding their features their goals and utility, but also regarding the super peer election criteria previous described. This section will focus each election process over it general concept, how the algorithm is designed, its focus points, the main characteristics and goals regarding the super peer election. The analysis of the covered existing election algorithms/systems on the literature is divided into two major groups: fixed and mobile election. An election with fixed nodes is analyzed using four approaches, and those are established by the order of complexity of the election process (Jim Dowling, 2006).

So for the fixed election mechanism and following the order of complexity, the first classification is the *Simple,* and is composed by systems were no specific election algorithm is defined, or exist a very simple and straight forward or even manual approach, to elect the super peers presented in the network. These were the first election algorithms used in a distributed *P2P* environment (derive from more traditional solutions), and were implemented with a set of very simple answers to an actually very complex problem.

The second classification consists of systems where the population of peers is divided into *Groups*, and super peers are elected within each group independently. The groups are usually based on peer properties such as physical location, network proximity, or semantic content.

The third classification contains systems where the super peer election method is based on the *Distributed Hash Table*. The *DHTs* are a well-known class of P2P systems, with a well-defined functionality, which constitutes a coherent subset of P2P systems. The advantage of DHT-based approaches over group-based approaches is that in DHT-based systems, peer clusters can be easily split and merged at runtime.

Due to the distinctive characteristics of the *DHTs*, systems that use *DHTs* for the election of super peers are considered as a separate class in this review (Papapetrou, 2008). Finally, the last classification contains *Adaptive* systems that elect super peers based on global demand, for example defined as the number of clients, rate of client requests, or current load on super peers. These systems usually define some optimization criteria and continuously strive to improve the super peer set. Following are described implementations that consider the election mechanism where super nodes in the network are mobile, and ad-hoc.

This type of network imposes restrictions on developed applications given its particular nature. Since the mobile super peer election does not present a large number of proposed solutions available in the literature, the mobile election was divided into two approaches: single and group election. In the end, a conclusion is drawn for the mobile election of super peers, and at the end of the chapter a summary is presented where are described all the considerations and conclusions drawn for the initial problem presented.

### 3.2.1 Fixed Approaches on Super Peer Election

Traditionally, *P2P* systems do not provide any algorithm for the election of super peers. Peers that participate in the system are selected manually for the super peer position, (through the use of additional implementations) by the local user at each peer, or by a global system operator that manages the network. Both approaches are used to provide an ideal and optimal super peer election in the network. The analysed fixed approaches are represented in Figure 3-1.

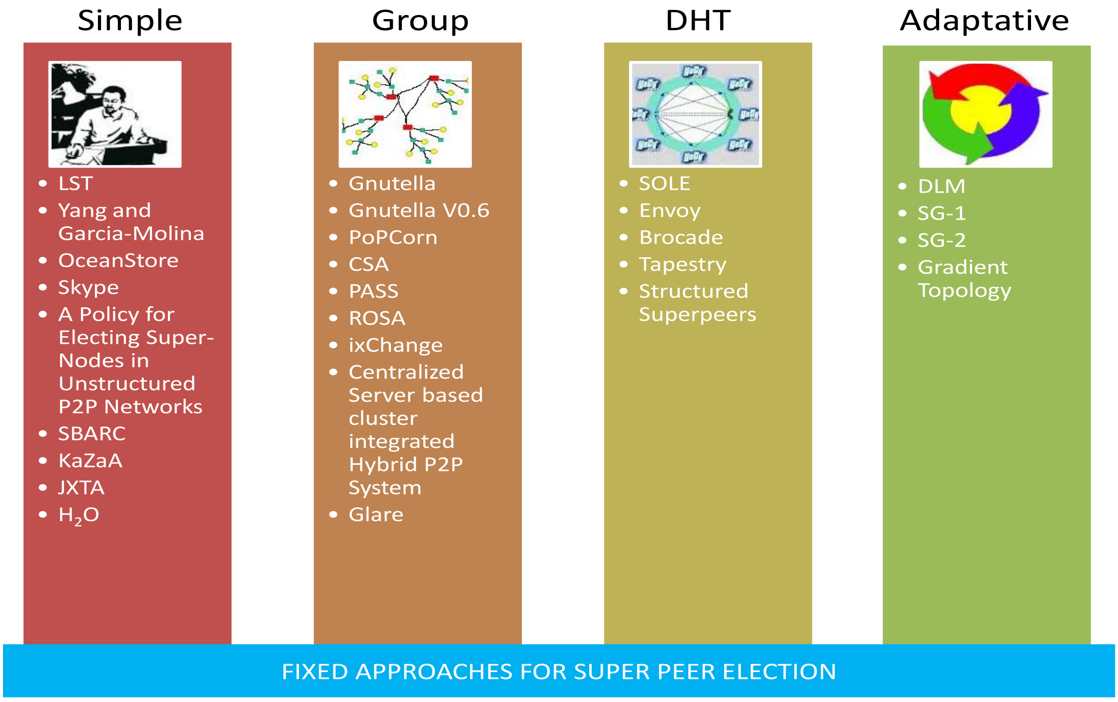


Figure 3-1 - Fixed Super Peer Election Methods covered by this review

#### 3.2.1.1 Simple Approaches

The *Lightweight Super Peer Topologies* (LST) proposed by *Kleis* (M. Kleis, 2005) is intended for routing in *P2P* networks (Lua, et al., 2007). However in the specification for the election approach it´s only mention that a "*SuperPeer should have sufficient resources to serve other SuperPeers and Peers", and that "The SuperPeer should be reliable and it is not joining and leaving the LST overlay network very frequently*", in (Eng Keong Lua, 2007). *Yang* and *Garcia-Molina* (Beverly Yang Hector, 2002) have only investigated the relationships between the number of super peers.

The load that the super peers can support, and search performance, in order to find the formula for an optimal system configuration, but they do not provide any specific super peer election algorithm. The *OceanStore*, described has a utility infrastructure (J. Kubiatowicz, 2000), proposed to elect a primary tier "*consisting of small number of replicas located in high-bandwidth, high connectivity regions of the network*" (Jan Sacha, 2006) for the main propose of handling updates. However, the specific algorithm for the election of such a tier is not present.

Another example is the *Skype* application where the protocol specification and source-code are not publicly available, but by reverse-engineering it's credible to say that peers are promoted to super peers if they are non-firewalled, have a high amount of bandwidth, and also based on a number of factors like CPU and available bandwidth (Schulzrinne, April 2006).

In "*A Policy for Electing Super-Nodes in Unstructured P2P Networks"*, (Georgios Pitsilis, 2004) the objective is to identify the best candidates and see whether they can be suggested as Super Peers, based on the generated traffic (number of messages). This process determines super peers candidates based on two connectivity measures: the outer degree and the elementary cycle value. The Outer Degree of node *i* indicates how many nodes are connected to *i*. Let $a_{ij}$ be the adjacency matrix where $a_{ij} = 1$, if "*i*" is connected to "*j*" and $a_{ij} = 0$ if "*i*" is not connected to "*j*". Let "*n*" be the total number of nodes. The Outer Degree is estimated by the following equation:

$$OuterDegree_i = \sum_{j=i}^{n} A_{ij} \qquad 3.1$$

Elementary cycle is a measure of the participation of all nodes of a network. To use the connectivity measures for the election, it´s required that the algorithm have a global view of the peers present and their established connections.

The *Supernode BAsed Routing and Caching* (SBARC), is a new a *P2P* file sharing system which takes into account the peers heterogeneity to improve the system performance. For a peer to be considered super, it needs to satisfy certain criteria, regarding a high network bandwidth, computation power, disk storage capacity, and that the peer can´t join or leave the system frequently. These criteria are directly connected to the system utility (file sharing), and no further election process or algorithm is revealed (Hu, 2003). *KaZaA* include the super peers to solve the scalability problems faced by Gnutella.

It is believed that the peer decides to become super, use local knowledge about their own characteristics, such as bandwidth, processing power, unrestricted access to the internet and availability, however it is not precisely known how the election mechanism works, since the source code is not openly available (J. Liang, April 2005), (Levene, 2010). The architecture of the *JXTA* (Juxtapose) network is similar to the one used in *KazaA*, (Bernard Traversat, 2002). For the JXTA protocol specification, any peer can be elected to provide the super peer service. The super peer election is usually made via the configurator tool (Scott Oaks, 2002), and it is assumed that any peer can become a super peer assuming that it has "*the right credentials*".

It is up to the higher-level application to decide on the minimum capacity and the necessary credentials to be elected to super peer. The protocol also established that after an established period, the peer will try to become super, if it has the right credentials (B. Traversat, 2003). The *Hierarchical 2-level Overlay* ($H_2O$) protocol for Super Peer Selection is a distributed negotiation protocol. With this protocol, all peers are autonomous to make a decision, and allowed to negotiate with each other's using its own local policy. For the negotiation, the peer advertises information about itself to its neighbors that includes trust level uptime, bandwidth, and neighborhood size. This information is used by the peer has part of the selection criteria to a peer become a super peer. It is also used the security certificates, since only peers that hold the certificate are eligible to be elected super peers, and the peer can declare itself has a super peers. But no reference to the negotiation process is mentioned (Virginia Lo, 2005).

#### 3.2.1.2 Group Approaches

In "*DIGITAL MEDIA ANALYSIS OF GNUTELLA PEER-TO-PEER NETWORKS*" (Heslep, 2006), the conducted analysis of the *Gnutella* protocol, establish the used super peer election principles, and the basic requirements that must be satisfied to a peer be considered for the super peer role.

For the *Gnutella V0.6* protocol, the election require that: the peer isn´t firewalled, the peer run over a suitable operation system (to handler the large number of sockets), has sufficient bandwidth (at least 15KB/s downstream and 10KB/s upstream) and uptime, sufficient RAM and CPU speed. If all those criteria are met, the peer is said to be a candidate to super peer, and now the peer can "estimate" it necessity, from the number of super peers present in the network, but no super peer election algorithm is mentioned (Anurag Singla, 2002).

The *PoPCorn* was designed as part of the *Cluster Computing On the Fly* (CCOF) project for *peer-to-peer* cycle sharing. PopCorn is suited for applications that wish to select a fixed set of "*k*" supernodes and distribute them evenly throughout the overlay. PoPCorns primary distribution criteria, is achieved by maximizing the sum of inter-node distances between all pairs of supernodes. The PoPCorn protocol selects "*k*" supernodes by dispersing "*k*" tokens through the overlay coordinate space using a repulsion model among the tokens. Each token represents one of the supernodes, which moves through the overlay based on the forces exerted on it by other tokens. The repulsion model is used to adjust the location of the tokens. When equilibrium is reached, each node holding a token is selected as supernode. (Virginia Lo, 2005).

*Cluster Server Architecture* (CSA) aims to realize an enhanced differentiated QoS in *P2P* file sharing systems. In each cluster, a powerful node is selected as the Cluster Server (CS) that serves for all others nodes (clients) in this cluster. In order to explore the heterogeneity among the peers, it has been established a candidacy factor to each node to be a CS candidate, described has the following function:

$$CS_{Candidacy} = f(NM, NB, CC, MC) \qquad 3.2$$

,where NM stands for node mobility, NB for network bandwidth, CC for CPU capability, and MC for memory capacity. These four factors carry different weights in the equation. To qualify for CS, a node must have a candidacy factor larger than a predefined threshold. (Ping Ge, 2008). *Peer-to-peer Asymmetric file Sharing System* (PASS) is a novel approach to *P2P* file sharing. This system selects only a portion of high-capacity machines (super nodes) for routing support. The special supernode that represents its area is called the Representative Super Node (RSN). The consistency of the RSN directory is maintained by using an epidemic protocol (Gisik Kwon, 2003).

The *ROSA* P2P system aims at providing a distributed environment which enables the communication between e-learning. To be elected super peer, the peer must provide some physical important characteristics has stability, bandwidth and fast access, but also processing, memory and storage capacity. The number of super peer present in each group is dynamic, and established according to the amount of peers that a super peer can support, and is parameterized according to an assessment of the response time for certain queries submitted over the machines (Gabriel André Duquesnois Dubois Brito, 2005).

The *iXChange* model utilizes the information available from participating peers to: cluster based on shared interest and intelligently elect high quality super peers. The user can decide is a peer is a super peer candidate or if the group requires a super peer election to take place. For election purposes, the peers share information in a number of categories: Available Bandwidth, System Memory, Processor speed, Shared Storage Available and Average Uptime. The algorithm takes one argument (a set of node IDs), as input and returns a node ID which is elected as leader (C. Mastroianni, 2005). Since each node independently runs the algorithm, and all the nodes share the same list of nodes, so they must reach the same result, and no additional communication messages are needed (S. Johnstone, 2005).

*Centralized Server based cluster integrated Hybrid P2P System* is a *P2P* system. Each node votes by sending a 'Reply' message to the supernode containing their willingness. This willingness holds nodes information about the speed of the CPU, the size of memory, bandwidth, and NAT connectivity, based on the new supernode. The election process is mandatory for all the nodes in the cluster (Rashmi Ranjan Rout, 2008). *GLARE* framework is self-managed and fault tolerant. One member from each group becomes a super peer and all super peers form a super group. For the election, the site with higher rank is elected as super peer. In order to rank the different sites, a unique hash code of all grid sites is calculated based on their static attributes. These attributes includes processor speed, memory, uptime and site name (M. Siddiqui, 2005).

#### 3.2.1.3 Distributed Hash Table Approaches

In the *Supernode Selection in Structured Overlay Networks* (SOLE) system, the super nodes are the nodes with better capability that respect to CPU speed, network connections, and other resources. Joining nodes can negotiate to see which one is more capable to take the supernode role. A nearby non-supernode can offer to take over a nearby supernodes role if it is more capable (Virginia Lo, 2005).

*Envoy* is a two-layer P2P network where a structured overlay is built on top of an unstructured one. In this system a peer sends some random walkers to search an existing super peer, or to recommend one. The super peers are elected based on their stability and capability scores, through mutual comparisons which ones are more qualified. The Stability of a peer 'x' then is measured by the total scores earned by x in some observed interval Δ, and for each measured session τ:

$$\text{Stability}_{\Delta}(x) = \sum_{\forall\tau\in\Delta} s(\tau) \qquad 3.3$$

Here the stability is defined has: $s(\tau) = (|\tau| - T)^{-log_{\alpha}k}$. Both scores can be weighted to reflect different application need. To avoid dealing with different distributions of scoring functions, the peer scores were normalized by defining a rank function for each peer (Yuh-Jzer Jounga, 2009).

The *Brocade* overlay architecture, improves routing efficiency in a DHT by exploiting resource heterogeneity, but it doesn't address the super peer election problem. The super peer selection criteria are that super peers have significant processing power (in order to route large amounts of overlay traffic), minimal number of IP hops to the wide-area network, and high bandwidth outgoing links.

The final choice of a super peer can be resolved by an election algorithm between Tapestry nodes with sufficient resources, or as a performance optimizing choice by the responsible Internet Service Provider (Ben Y. Zhao, 2002). The *Tapestry* approach is a new super peer election algorithm based upon unstructured network. Since each peer has many different capabilities: processing ability, storage capacity, connective ability, uptime, etc. All the integrated factors give the following function:

$$G = \frac{MTBO}{MTBO_S} x \frac{CPU}{CPU_S} x \frac{V}{V_S} \qquad 3.4$$

Where MTBO was the mean time between offline, MTBOs was the minimum MTBO as a SuperNode, CPU meant the effective CPU processing ability and CPUs was the minimum processing ability as a SuperNode, V meant the effective connective speed and Vs was the minimum connective speed as a SuperNode.

With the exchange of information, every node gave a vote to the best node according to its computing result, and the node with the maximum votes, would be elected Super Node.

The main contribution of this algorithm was by introducing the distributed and parallel election process into the SuperNode election, which might raise the election speed and reduce message complexity of the election algorithm from $O(n^2)$ to $O(n^2/k)$ ,where n and k were the number of nodes and districts respectively (Cuibo Yu, 2009).

The "*Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup*" described in (Alper Tugay Mızrak, 2003) exploits the resource heterogeneity inherent in existing *P2P* systems. It is assumed that each super peer knows its maximum capacity and measures the current load. To tune the load balancing behaviour the approach defines four limit parameters: *min, max, lower, and upper*. The first two represent hard limits on the capacity of the super peer, while the latter two are soft limits meant to initiate load balancing activities long before a super peer is overwhelmed or idle. While the size of each message is constant, the number of messages to be sent grows linearly with the number of superpeers.

### 3.2.1.4 Adaptive Election

The *Dynamic Layer Management* (DLM) algorithm is specific to file-sharing applications. Ideally, the superpeers should be more powerful and with a longer lifetime than average peers. To measure the eligibility of a peer, the authors define two metrics, Capacity and Age (Li Xiao, 2005). Since it and cannot be easily applied to other areas it doesn't involve the use of external conditional decision. The optimum number of super peers in the system is calculated based on estimated system properties and that requires the knowledge of the average duration of peer connections, which may depend on the system deployment environment, and hence only can be obtained at runtime.

*SG-1* is a protocol for the construction and management of the super peer based overlay topologies. The protocol is based on the gossip paradigm, to build a topology characterized by a minimum number of super peers. The overall communication cost of the algorithm is given by the total amount of messages exchanged by the multiple layers of the system, and there are only two exchanges per node (Montresor, 2004). SG-1 has the drawback that it does not allow the system to explicitly control the number of super peers and to adapt the super peers set to the current total demand in the system.

Also it is necessary to frequently exchange messages with the neighbours that increases the overhead related to the overlay network maintenance. The novel protocol for building and maintaining proximity-aware superpeer topologies is named *SG-2*. This protocol also uses the gossip-based protocol to spread messages to nearby nodes and implements a biology-inspired task allocation protocol that mimics the behavior of social insects to promote the "best" nodes to super peer status. It used the 'θ' adaptation process that is only active when a node is in the client state. Furthermore, a global parameter '*tol*' expresses the maximum latency distance that can be tolerated between clients and superpeers. The latency distances that are "under-estimated" may pose a problem: if the actual latency is over '*tol*', but the estimated latency is smaller, a super peer may accept a client out of the tolerated range.

For this reason, the maximum error must be considered when selecting parameter '*tol*'. (G. P. Jesi, 2006). The problem of finding the target topology, even in a static system, is NP[3]-hard (J. Liang, April 2005). So in a dynamic environment, with joining and leaving peers and with communication failures, the problem is even more difficult. For the *Gradient topology*, the highest utility peers (servers), maintaining persistent data, are highly connected with each other and form a logical core of the network, while the network around the core is composed of other peers (clients) considered less performance or less reliable. The authors describe that the algorithm, relies on the existence of a utility function that captures the peer application specific constraints. A utility value above a certain threshold makes a node eligible for super peer role. The peers at the core (also named super peers) should be well-connected, have high bandwidth and processing power, and should be able to maintain a relatively high number of connections (Dowling, 2005) (Jan Sacha, 2006) (Jim Dowling, 2006).

A decentralised aggregation technique allows peers to estimate the distribution of peer utility in the system and to identify an adaptive super peer selection threshold.

[3] The complexity class of decision problems that is intrinsically harder than those that can be solved by a nondeterministic Turing machine in polynomial time. When a decision version of a combinatorial optimization problem is proved to belong to the class of *NP-complete* problems, then the optimization version is NP-hard.

### 3.2.1.5 Conclusion

For the simple approaches the *Yang and Garcia-Molina*, *JXTA*, *PoPCorn and GLARE* uses preconfigured values as the thresholds to select super peers, which lead to a position which is impossible to maintain an appropriate size ratio on the network. The *CSA* adopts the approach to use factors that carry different weights in order to provide the algorithm with conditional decision but there is no information on the possibility to change the weights. Envoy also used weights and the measured peers' scores were normalized. This list of nodes used by *iXChange* is the first approach to the local knowledge described in the proposed challenges. It also intends to minimize the number of exchanged messages, but it is limited to use the node ID for the super peer election.

For the *SOLE* system, there is a reference that the super peers should have sufficient capability regarding the CPU speed, network connections and other resources. *Structured Superpeers* exploits the resource heterogeneity, but it also imposes limits parameters, and with large-scale networks the number of messages necessary to the super peer load balancing behavior also grows. *DLM* uses relatively simple heuristics for the estimation of global system properties, but the description of *DLM* is missing details. In particular, it does not explain how the threshold parameters, $Z_{c,p}$ and $Z_{a,p}$, and scale parameters, $X_{c,p}$ and $X_{a,p}$, are calculated. In *A Policy for Electing Super-Nodes in Unstructured P2P Networks, SBARC, $H_2O$* and *Tapestry*, the goal is to identify the best candidates based on some establish parameter. For *A Policy for Electing Super-Nodes in Unstructured P2P Networks* is also necessary to have a global view of the peers and the established connections. Those approaches preclude the use of these systems out of context for which they were implemented and cannot be easily tailored to a wide variety of applications.

SG-2 uses *Spherecast* and it is not clear if the *Spherecast* algorithm scales when the density of peers in the system grows, and the number of messages. Also if a low-performance peer is located in an area with a high density of other peers, it may easily become overloaded. Thus, the range of the influence zones is a critical factor affecting the systems performance that may not have a trivial solution. The emergence of a gradient topology is a result of the system's self-organisation. Peers are independent, have limited knowledge about the system and interact with a limited number of neighbours.

In the *MANET*s, the peers are free to move, regarding that should exist a limited knowledge about the system, and since the number of neighbors may be in constant changes, the Gradient topology will in most cases never reach a steady state of self-organisation. The use of gossip protocols (PASS) here the neighbourhood is chosen randomly was proven to work efficiently were the bandwidth is abundant and communications between faraway nodes is not constrained by the scarcity of physical resources. Nevertheless, their implementation in wireless networks is unknown. In this field, there is research that aims to analyze this implementation, as described in **"***A Comparison of epidemic algorithms in wireless sensor networks"* (Mert Akdere, 2006), that for some cases it states "*However, the contention in the network and the power consumption also increase with increasing transmission range. As a result, transmission range should be optimized with respect to the application scenarios*".

This random neighborhood selection policy has been defined as not suitable for wireless ad hoc networks where communications between faraway nodes incur a high routing overhead, which impairs the performance of the systems. To use most of the described systems on top of a mobile network the result would be inefficient, since the algorithms will cause extra network overhead, because the election scheme, does not consider the dynamic changes the user behavior and the mobility in the network for the election. For other approaches, their performance has only been evaluated in Fixed/Static environment and retains unknown the results when the nodes are moving.

All methods covered by this review have shown to possess distinctive features, in their mechanism, but while many of them are not access to the public (*LST, OceanStore, Skype, KaZaA*), others are incomplete and doesn't provide and effective election mechanism (*Gnutella, Gnutella V0.6, ROSA, Centralized Server based cluster integrated Hybrid P2P System, Brocade*). From the analysed approaches no information exists regarding the proposed challenges described earlier and in others system the election only used the node information and no distributed decision making is performed or even messages are exchanged. Many of those elections mechanisms do not take into account the restrictions/constraints imposed by wireless networks. Particularly regarding the high messaging, the availability and misuse of bandwidth, and considerations over the use of batteries. All those factors must be taken into account when the algorithm is intended to be used in *MANET*s. The mechanisms for fixed election approach does not take into account the mobility associated with wireless networks, or even the high dynamism that *MANET*s present, and any of them presents itself has a solution for the election in *MANET*s.

## 3.2.2 Mobile Approaches on Super Peer Election

In the election mechanism where the nodes are mobile, beyond the concern regarding an distributed election, must also be taken into account the inherent mobility to the nodes in the network. At this section all the covered election mechanism are suitable for *MANET*s, and are analyzed according with the problematic of "***keeping the super peers topology over a MANET***". The analyzed mobile election mechanisms (Figure 3-2) are analyzed individually, followed by a review of the existing mobile super peer election solutions in the literature.

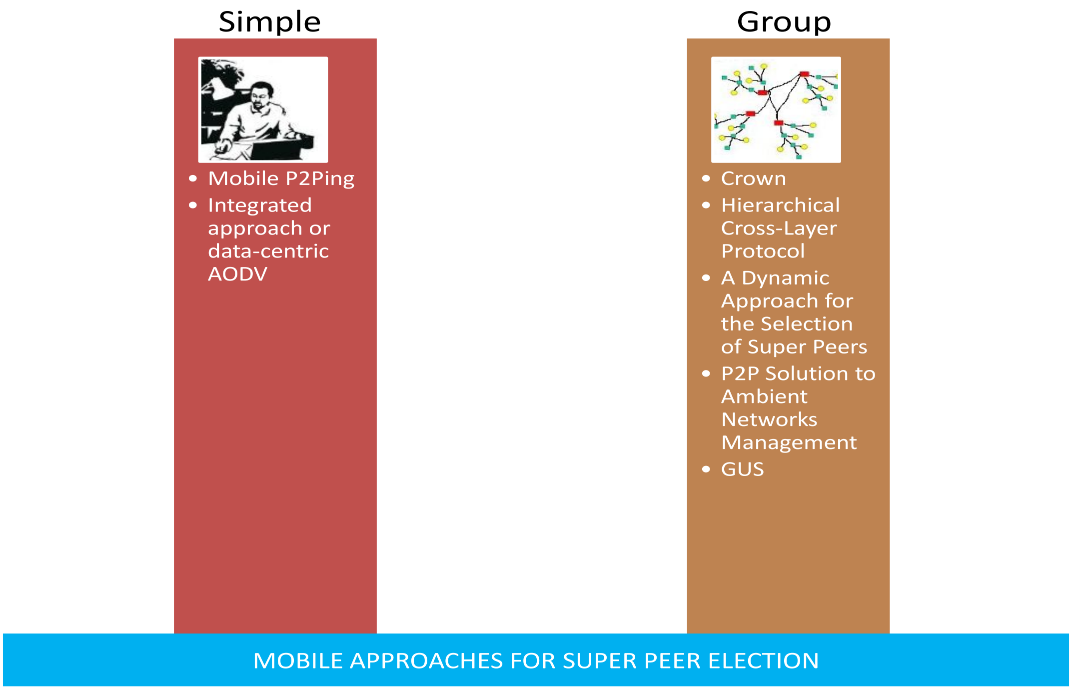


Figure 3-2 - Mobile Super Peer Election Methods covered by this review

### 3.2.2.1 Simple Approaches

The *Mobile P2Ping*, is a special case were the mobile super peer system is applied to the city buses. The MANET is based on mobile fleets such as city buses, that can be wireless-enabled. Those specific buses are able to inter-communicate and serve as mobile routers for other nodes connected to them, and the extensive bus network with services that links to virtually every corner of the city.

For this particular case the buses are elected has super peers by the system administrators, and serve as super peers (Seet, 2004). Another approach is the *Integrated approach or data-centric AODV,* which is a common query/response framework in which *P2P* file sharing and Ad Hoc environment are integrated seamlessly. It used the AODV for ad hoc routing protocol integrated with *FASTTRACK P2P* file sharing protocol. The super peer implementation is very simplified, since the super peer election technique were not implemented, and the super nodes were placed symmetrically inside the network, to cover each region of a virtual grid (Bin Tang, 2005).

#### 3.2.2.2 Group Approaches

The Crown protocol is suitable for mobile and wireless network environment is made up of many groups, which are mutually connected into a ring (Figure 3-3). The protocol uses the Super Peer Election Mechanism to elect super peers. The super peer election mechanism is based on five parameters: Bandwidth and Work Load, Availability, Computation Power, Node Characteristic and finally with User Intervention. Regarding the election mechanism no more is described. (T.I. Wang, 2004).

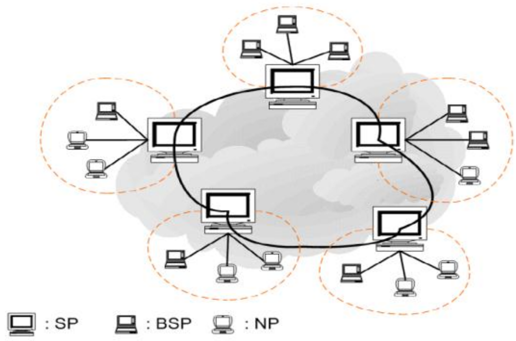


Figure 3-3 – Crown Topology

A *Hierarchical Cross-Layer Protocol* that uses cross-layer optimization has proposed for Group Communication in *MANETs*, which combines both gossip-based and tree-based schemes for data dissemination. It used the peer Capability (measurement of CPU process ability and memory storage, which is predefined and never changes) has a selection measure to the role of Super Peer. For the Super Peer election, the protocol establishes global defined parameters, which are to elect normal peer to the super peer state.

Those parameters are: *minCap* (Minimum capability for a node to be able to take the responsibility of Super Peer); *maxDis* (Maximum distance from normal peer to its connected Super peer); *avgCap* (is the minimum average capability that an Super Peer should at least allocate for each of its connecting normal peer). Those are global and static parameters that each node has to fulfil in order to be elected super peer. With this mechanism no further negotiation or additional information is needed for the election (Yifen WEI, 2007).

For the *A Dynamic Approach for the Selection of Super Peers*, the super peers are selected based on some rigorous criteria. Every node is randomly assigned a superiority ratio r, a measure of node capabilities such as computational power and battery life. A node is a super peer candidate if its superiority ratio exceeds a certain threshold h. It was implemented two selection approaches: Control Loop and Event Triggered. The authors use five metrics to evaluate the impact of the selection process on the network; Mean Number of Switches, Mean Time between Successive Switches, Non-Candidate Percentage, Maximum Service Time Percentage, and Mean Number of Hops (Ahmed M. Mahdy, 2007). The *P2P solution to Ambient Networks Management* project is a distributed network management framework. In order to negotiate with other peer groups, each peer group elects the representative super peer, for each peer group. The super peer management information is distributed within the group.

The project describes that the "*management application have also been implemented including the super peer election module, the network composition logic and the policy negotiation and maintenance modules*" (R. Szabo et al., 2005). However no further description or reference is described in the consulted document regarding the super peer election policy (Csaba Simon, 2005). *Gnutella Ultrapeer System* (GUS) was proposed to improve the performance of the Mobile Collaborative Virtual Environments. The elected super peer is defined has a peer with the following criteria: powerful mobile device having higher computational power, storage capacity, and network bandwidth availability.

Each node can determine itself to be super peer using an on/off flag. The switching process between node statuses is triggered whenever an improvement in the GUS is required. The *Lowest-ID algorithm (LAC)* is the most simples, popular and used algorithm to face the cluster head election (Tsai, 1995). Nodes periodically broadcast their ID, after each node compares the IDs of its neighbors with its own ID, and the node having the lowest ID decides to become the cluster head (Azzedine Boukerche, 2009).

### 3.2.2.3 Conclusion

Regarding the *Mobile P2Ping*, the super peer are the city buses, and elected by the global administrator, and no messages or election mechanism is necessary, since the administrator takes all the decisions. *Integrated approach or data-centric AODV* that uses a virtual grid simplifies the election by placing the super peer symmetrically inside the network, and to cover each region. This requires a global view of the network, which is not combative with scalability. Other mobile super peer approaches (*Crown*) use fixed parameters for the super peer election, but also provides the possibility to the uses elect the desired super peer.

*The Hierarchical Cross-Layer Protocol* intends to minimize the number of exchanged message thought the use of global and static parameters. But those parameters present a serious drawback for the election. The static threshold does not allow the system to control the number of super peers in dynamic populations of peers. If peer properties change, the super peer sets changes accordingly. This can lead to extreme cases, where no super peers are elected, if all peers fall below the threshold, or where every peer is a super peer, if all peers are above the threshold. *P2P solution to Ambient Networks Management* suited for mobile networks, and used a distribute decision-making ability to elect the super peers.

Each peer negotiate within it group however it is mention an election algorithm, no policy is presented. *A Dynamic Approach for the Selection of Super Peers,* takes into consideration the battery life, but still uses a defined threshold that is used to maintain a defined ratio between the super peers present in the network and to be used into the election mechanism to elect new super peers. The *GUS* does not take into account the peers heterogeneity in the network, and only focuses on the election of a peer for each cluster.

It uses the distinguishing characteristics of the peers, as the Peer ID to elect the super peers, and the super peer state is maintained by the use of a flag, affected by the Lowest-ID algorithm. But the approaches integrated into the group approach if the total number of peers in these systems is low, groups have relatively few peers, and the overall ratio of clients to super peers is low. Contrarily, if the number of peers is large, groups consist of large numbers of peers, and the ratio of clients to super peers is high. For a certain system size, super peers will become overloaded. Inherently, such a system does not scale. Also if all super peers were clustered in the same region, there would be too many messages through that part network which might result in bandwidth crowded and decrease the network efficiency.

Many approaches often elect the “high utility super peer” based on some “pre-defined criteria’s”, has the higher computational power, higher storage capacity and network bandwidth amongst others. But this has a huge drawback because before a peer attempts to discover a high utility peer, the peer needs to estimate the distribution of peer utility in the system in order to know what constitutes high utility in a running system. This often provides an inadequate election, driven by the distribution of a peer utility for the entire system.

Some approaches take into account the number of messages exchanged in order to not excess the bandwidth, and also not to consume the batteries of the devices, others take into account the mobility of peers in the network. In (Emna Salhi, 2009), the study performed on mobile P2P states that the best performance is archived by limiting the neighbourhood scope to small values without any diversification effort. In rapidly changing network environments with relatively limited-bandwidth, dynamic negotiations for Super Peer election, and subsequently re-election are not desirable due to the increased overhead they present. Thus, high node mobility would potentially, result in frequent Super Peer re-election. Nevertheless, none of the analyzed approaches taken into consideration all the election criterions, or provide an effective and applicable election mechanism that is independent of the system utility.

## 3.3 Summary

For *Gnutella, Gnutella V0.6, ROSA, Centralized Server based cluster integrated Hybrid P2P System, Brocade* election algorithms are mentioned, but no details of such an algorithm are provided.

Approaches as *Envoy* for the super peer election based on random walking, are difficult in large *P2P* systems due to high communication overhead that increases as the size of the *P2P* system grows. Manual or static election is not likely to produce optimal super peer sets due to the complexity and dynamism in most *P2P* networks. In (Jim Dowling, 2006) is described that “*Solutions based on manual or static configuration of super-peers are inappropriate due to a lack of system-wide knowledge of peer properties*” For the *CSA*, *Yang and Garcia-Molina*, *JXTA*, *PoPCorn, GLARE, DLM, A Dynamic Approach for the Selection of Super Peers, Hierarchical Cross-Layer Protocol* and *Gradient topology* the peers that satisfy certain requirements, are selected to be super peer by some authority or by fixed thresholds.

Fixed thresholds can only be applied to networks where the distribution of system-wide peer characteristics does not change significantly in time and is well known by the system designer or administrator. Those mechanisms can be classified as incomplete methods.

The groups approaches (*Gnutella, Gnutella V0.6, PoPCorn, CSA, ROSA, iXChange, Centralized Server based cluster integrated Hybrid P2P System*, *GUS and SOLE*) have a number of drawbacks. First of all, they do not allow the system to actively control and adapt the number of super peers to changing system conditions.

The number of super peers is determined by the current number of groups and cannot be tuned at runtime. Furthermore, in many systems, the maximum number of groups is fixed. While multiple super peers can be elected in each group, however, this requires additional mechanisms for the synchronisation and coordination of multiple super peers elected in one group, and dividing clients between super peers in a group.

The adoption of additional synchronisation mechanisms may lead to the use of additional messages in the system. Ultimately, as the system size grows to a large number of peers, the problem of multiple super peer election in a group becomes equivalent to the general problem of super peer election in a *P2P* system. In others systems, such as Crown, PASS, the numbers of super peers is not strictly controlled and depends on external factors such as peer IP addresses (Crown), peer locations (PASS), and local users or administrators. Furthermore, some systems introduce additional constraints on super peers and clients, such as a maximum distance in a virtual coordinate system or semantic space and it's also necessary to know the exact number of existing super peers in the group, in order to choose its best peers.

Due to the constraints imposed by the *DHT*, the system cannot at the same time elect the highest capability super peers and distribute clients evenly between the super peers. Furthermore, if the clients are not evenly balanced between the super peers, the total number of super peers in the system cannot be minimized. For example, if the best candidates for super peers have close *DHT* identifiers, which corresponds to the situation where high-capability peers are located in one group in a group-based approaches, the system is forced to choose between electing the highest-capability super peers, and electing the lower capability super peers that evenly divide the *DHT* space. Such a trade-off is particularly likely if the *DHT* identifiers are not purely random, but are rather generated based on the peer properties, such as peer location.

Also many of these systems only focus on load balancing in a *DHT* rather than the election of potential super peers from the set of all peers in the system.

Also due to the constraints imposed by the DHT, these systems cannot guarantee that the elected super peer sets are optimal in terms of size and super peer capabilities. The enhanced robustness of *Group*´s and *DHT*´s approaches is obtained at the expense of wasting bandwidth resources, which is quite limited in *MANET*s. None of the simpler, group or *DHT* approaches uses a mechanism that can adapt and be effective at the same time when the network purpose changes, since most of them have hardwired thresholds or are linked to the system utility or they are missing details in the election algorithm.

So the pre-defined criterions (minimize the use of exchanged messages; use of local knowledge; a distributed decision making ability; use of external conditional decision) that the algorithm has to support are inexistent, or only one or two are supported at the same time. The use of gossip can be much more reliable due to their inherent redundancy, but suffer from unbearable bandwidth overhead especially when the group size is large, and thus lack of scalability.

In summary, the majority of existing systems offer simple and limited mechanisms for the super peer election and only a handful of systems attempt to optimize the super peer sets according to well-define criteria. For most of them, the methods involve high message passing overhead, and many of them are not designed for large scale *peer-to-peer* networks that exhibit a high degree of churn and that are dynamically heterogeneous. Also these few sophisticated systems are only specific to particular application scenarios and a more general approach is still needed.

To conclude the reviewed systems offer relatively simple super peer election solutions, and there is a general need for more sophisticated techniques. Those solutions would allow peers to better control the number of super peers in the system and to adapt the super peer set according to changings over the system conditions.

# CHAPTER 4 Lightweight Distributed Super Peer Election Algorithm

This chapter presents the definition and design of the election mechanism, aiming to response the Super Peer Election Problem according with the set of predefined super peer election criteria. Those criteria are according to use the minimal number of exchanged messages, associated with the use of the nodes local knowledge and moreover the ability to support distributed decision making and also external conditional decision. To establish an election mechanism to deal with the properties of a *MANET*, the algorithm uses the local knowledge, available from the surrounding peers (neighbours), to strengthen the algorithm and to provide the election with independence in face of the network size. This approach gives the algorithm its scalability and flexibility, since the increase number of peers in the network does not in any way add complexity to the algorithm. The use of a distributed decision making approach ensures that the final outcome of the algorithm take into account the neighbouring nodes, and use information on these as the main source for election results.

## 4.1 Definition

The main focus of this work is related to the election phase, and is established that the eligibility of a peer to be set as super or edge has already taken place. The peers that are in this stage receive the name of Super Peer Candidate (*SPC*), since they have all the requirements to be elected super peer, and only are candidates for the super peer election. All the others peers that not considered SPC, are non-Super Peer Candidates. After setting the starting point, the first action is to combine all the possible combinations of the super peers that are present on the network. After setting the combination the "chosen one" will address the peer to be elected was Super Peer. The algorithm is described in the following steps.

***Step 1****: SPCs sends information for their neighbours to have knowledge of the SPCs*

From the point of view of the SPC, its neighbour peers consist of a limited set of peers that have established a direct logical link to the SPC. This link between the SPC and a neighbour is considered as one hop. The others peers connected to one SPC neighbour are two hops away from the SPC. So as the number of the maximum parameterized hop increases, the SPC can send its information to a larger number of peers.

In this approach, the SPC will send its information to all the peers which are a maximum of two hops away. Sending information just for two hops distance permits each peer to inform only its current nearby neighbours. This allows the neighbourhood to collect and to storage the local information they received. As this information is collected by the other peers in the network, they can structure it and build a local view of the network. This local view of the network will have information about the SPCs that are in a range of two hops. So, for peer "A" its neighbours at one hop distance are represented by SPC (A, 1): Neighbours = {B, C}. At a distance of two hops, are all the other peers connected to each neighbor of the SPC. The representation is described by SPC (A, 2): Neighbours = {D, E}. The maximum hop value can be adjusted. This process is shown in Figure 4-1.

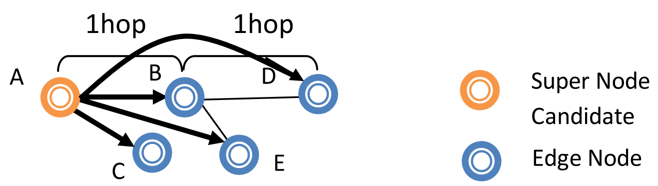


Figure 4-1 - SPC exchange information with its neighbours for two hops distance.

Each node will gather information about the SPCs in the surrounding network, so each SPC will send its *Status Table* (Figure 4-1) to all the other peers in the two hops range. The Status Table has information about the *Node Name*, the number of hops that the message stays in the network (*TTL*), the node parameters and the *Node View*. The node view is formed by all the neighbours of the SPC which are two hops away. For each neighbour it has the Neighbour Name, the Latency[4] to it, the last time the table was sent, and the number of hops to the SPC. The Status Table is send according to a random value, calculated internally in each peer.

[4] The time from the source peer sending a packet and to the destination peer receiving it.

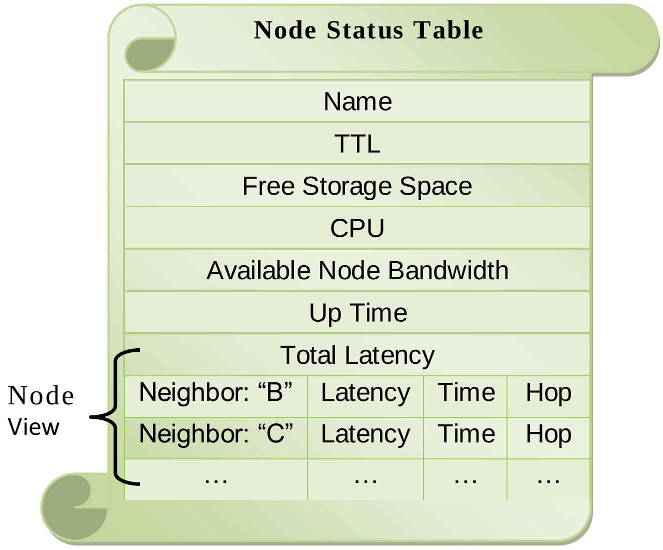


Figure 4-2 - The Status Table of a Peer sent for two hops distance

***Step 2****: Creation of a Utility table*

On this step, each peer creates its utility table. Since the SPC sends its status table to its neighbours at two hops distance, they all will have information about the SPCs on the network. With this information, the utility table will have all the possible combinations between the known SPCs that are at the range of two hops. Each combination can be seen as a possible option for electing the super peer. For each combination, the number of reachable peers for the current peer at a given hop will be evaluated. If a peer is reachable at one hop, it will never be taken into account at any other hop, so a peer is only reachable at the lower hop.

Following the information from Figure 4-1, all the possible combinations for the peer “B” are in Table 4-1. At this peer, there will be only two possible combinations since there is only one SPC in the range of two hops. It will then be possible to elect the peer “A” or not. After that, for each combination, the number of reachable peers for peer “B” is calculated at the selected hop. The utility table is shown in Table 4-1.

One network with several interconnected peers will only potentiate an increase in the number of combinations that are calculated on each peer. Since the method uses the current number of connections on a peer, it allows the algorithm a great capacity for adaptation to the network topology. If the network topology undergoes a radical change, this method has great agility and adaptability, since it requires no global knowledge, nor uses static thresholds for the election, which would need to be recalculated to collect information for all peers present.

| Node B | | | |
|---|---|---|---|
| | **Possible Combinations (*i*)** | **1 Hop** | **2 Hops** |
| **1** | **None** | [D,E,A] | [C] |
| **2** | **Super Node Candidate A** | [D,E,A,C] | [-] |

Table 4-1 - Utility Table regarding peer B.

***Step 3**: Elect the best SPC based on different criteria: reachable nodes, the rating, the future neighbours and the capacity*

In this third step, the best combination of all the possible combinations is selected from the utility table. Each combination will sum all the reachable peers from one hop, to two hops. This sum represents all the reachable peers that the selected combination offers.

After this calculation, the combination that offers the higher number of reachable peers will be selected. If two or more combinations have an equal number of reachable peers, there will be a rating factor, which will focus on the nearest hops and harm hops further away. The rating is computed in the 4.1 expression.

$$N\ Rating\ SPC\ (A,N) = \frac{\sum_{n=1}^{N} [((N+1)-n)*|SPC\ (A,n)|]}{\sum_{n=1}^{N} n} \qquad 4.1$$

$$;\mathrm{n} = \text{current number of hop.}$$

Figure 4-3 - Equation to calculate the Rating

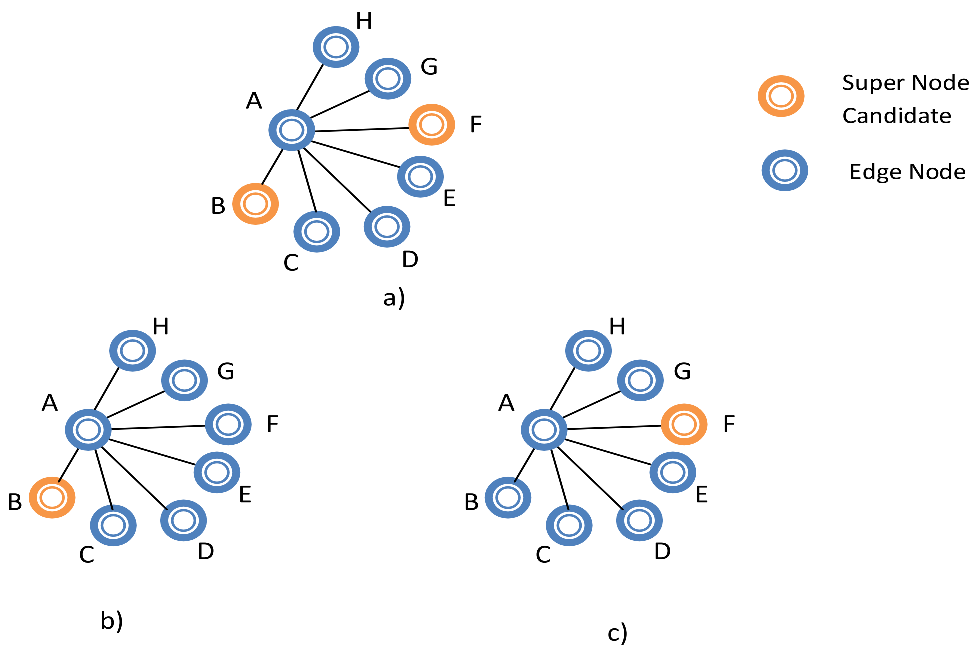


Figure 4-4 – Network in Star Topology with Super Peer Candidates. a) Combination 1: Network with super peers B and F; b) Combination 2: with only B as Super Peer; c) Combination 3: with only F as Super Peer

In some topologies the previous approaches will not be enough, since for some combinations the number of reachable peer for each hop is the same, and no advantage is obtained from the combination. One example is shown in Figure 4-4, where for peer A the rating for each combination (Figure 4-4 - b) and c) is the same. With the combination on Figure 4-4 - b), peer "A" can reach 7 peers with 1 hop, and 0 peers with 2 hops, and the same occurs in the same case in the combination represented in Figure 4-4 - c). With this particular network topology, the SPCs are the peer B and peer F, but exists also the possible to establish a combination using both of them. The resulting values are registered in Table 4-2.

| Node A | | | | | |
|---|---|---|---|---|---|
| | Possible Combinations (*i*) | 1 Hop | 2 Hops | Total nº of Reachable Nodes | Rating |
| **1** | **None** | [B,C,D,E,F,G,H] | [0] | 7 | 4,(6) |
| **2** | **Super Node B** | [B,C,D,E,F,G,H] | [0] | 7 | 4,(6) |
| **3** | **Super Node F** | [B,C,D,E,F,G,H] | [0] | 7 | 4,(6) |
| **4** | **Super Node B and F** | [B,C,D,E,F,G,H] | [0] | 7 | 4,(6) |

Table 4-2 - Utility Table for peer A

For Figure 4-4, the utility table of node A is represented in the Table 4-2. As it can be seen, all the possible combinations have reached a tie in the corresponded rating for each one. This tie is mostly due to the topology of this network, associated with the number of peer connections.

In the case of a rating tie, the combination that uses the minimum number of super peers is preferable to any other, since that the network will have the same behaviour with the least number of super peers. If the tie still occurs between different super peer combinations, the privilege combination will be the one that reach the highest number of neighbour peers. This is accomplished by adding the neighbours of each peer present in the combination. The result of this sum reflects the number of reachable peers by a combination. With the higher number of reachable peers, the chosen combination contributes to the election that favours the peers that have a greater number of neighbours, and consequently they are positioned on the denser areas of the network.


Figure 4-5 - Square network topology

The last decisive criterion will be to retrieve the best combination and the correspondent super peer elected on each non-SPC within one hop range. The peer will receive from each non-SPC neighbour their elected super peer. The peer with most recommendations will be selected. In case of a tie, since all the criterions have already been evaluated, all those that are tied will be elected super peers. For the example shown in Figure 4-5, the square network topology, the criterions used are in Table 4-3 that shown the criteria applied to each possible combination.

For the square network topology, the best choice will be the peer which has the highest capacity between SPC “A” and “E”, since these are the remaining combinations. If the capacities of peer “A” and peer “E” are the same, there will be a tie and both will be elect super peers. The measurement of the capacity parameter isn´t addressed in this thesis. This value results from the use of external solutions to this thesis and more appropriate developed for the calculation of this parameter.

| Node B | | | | | |
|---|---|---|---|---|---|
| | Possible Combinations (*i*) | 1 Hop | 2 Hops | Rating | Capacity |
| 1 | None | [A,C,E] | [D] | 2,(3) | 0 |
| 2 | Super Node A | [A,C,E,D] | [-] | 2,(6) | $C_A$ |
| 3 | Super Node E | [A,C,E,D] | [-] | 2,(6) | $C_E$ |
| 4 | Super Node A and E | [A,C,E,D] | [-] | 2,(6) | $(C_A+ C_A)/2$ |

Table 4-3 - Table with the criteria applied to each possible combination

This algorithm started when the node establishes a connection with another node that already exists (when the node performs the operation "Join"). The algorithm is them invoked again after a random time.

***Step 4:*** *Ad Hoc Module*

Besides those internal criterions already described, it's also possible to implement different/new criterions in parallel to influence the election decision. With the addition of an *ad hoc* criteria module is possible to change the final result without any alteration on the already established criteria. With this module is possible to establish an external control variable over the algorithm that driven according to the super peer election preferences.

***Step 5****: Synchronization of the elected Super Peer*

In some cases, the best solution might consist of two nodes, all the nodes, or even no Super Nodes. After finding the best solution from the combinations of SNCs, the node sends one election message and the SNC that receives it becomes the Super Node. At the end of the process, all the nodes that have received one election message will be elected Super Nodes and a synchronisation step will stabilize the final number of Super Nodes.

This message will also have an associated TTL, which must be forwarded in the network to prevent loop situations. The message is sent with the source name of the node, and the destiny node that will become a SNC.

| Message | | | |
|---|---|---|---|
| Message Name | Source Node | SNC Node | TTL |
| “Super_Node_Request” | Name | SNC | Maximum number of hops |

Figure 4-6 - Super Peer Election Message

## 4.2 Specification

The developed approach makes use of set of parameters to elect the super peer. The evaluated solution has to allow the algorithm to hold an election in each peer independently and without the need for timings, or any other dependencies. The algorithm specification is described below.

The neighbourhood selection policy is considered the first building block for the super peer election. So initially each time that a peer enters the network, it send a "*JOIN*" message to establish a connection with the peers already present in the network. After the connection is established each peer must create a list, which will contain all the establishing links to the other nodes (neighbours). The list is populated with all the connections established between the peers and corresponds to the “*node view*” which was previously described. This list is modified during the entry and exit of peers in the network. This list reflects the established links with other peers in the network in real time, and it’s part of a larger structure named “*utility table*”.

The utility table consists of the node view and the fields described previously: *Node Name and TTL.* It is worth mentioning that for this step there will be only considered the already eligible nodes to take the role of super peer. The previously created list, with all the established connections, will be sent to the neighbouring peers present in the network with the function: “*RandomsendMsgUtilityToAll ()*”.

This process thus enables the exchange of information that is only present on each peer to serve as ground for a wider knowledge about the network defined has local knowledge. The list will be sent with a *TTL* = 2, to ensure that it only reach the peers who are at a distance of 2 hops. This local knowledge is then obtained through multiple exchanged utility tables, from peers that are at the distance of 2 hops.

After the amendment or modification of the node view (peer join or leave), this list will be sent to neighbouring peers, so that those may also have updated information on the existence of new peers in the network. Each received message that contains the node view, is stored using the function:"*storage_msg (msg)*". So the entire process, since the entrance of a peer in the network, the exchanged messages between peers and the spread of the node view is described in Figure 4-7.

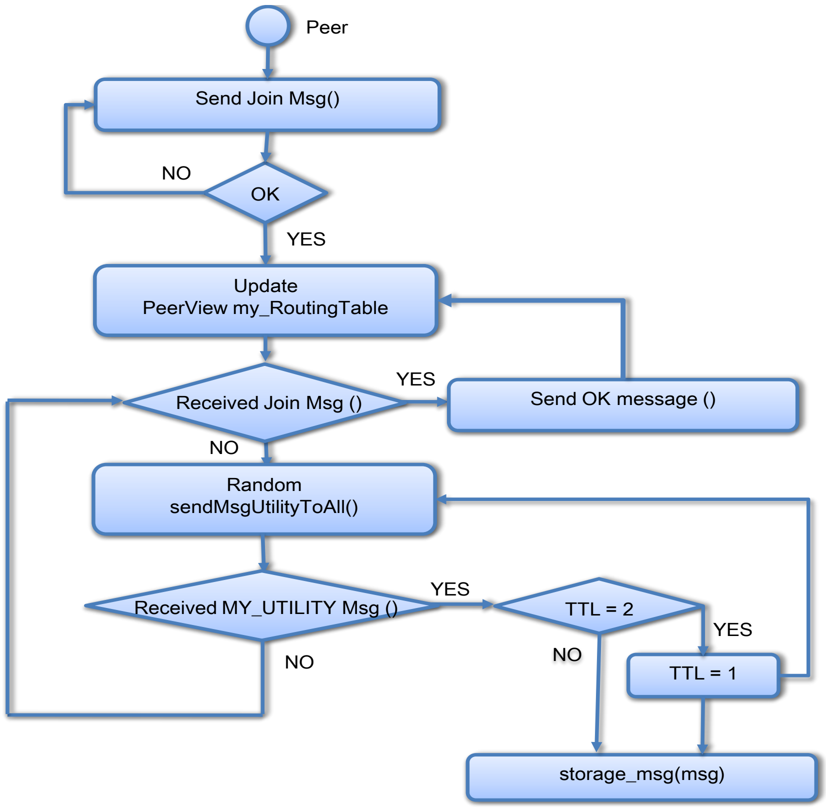


Figure 4-7 - Diagram: Collect peer information and Storage

Once the messages are stored at the peer, it's necessary to filter the ones that are arriving from the exactly same peer. This may happen if a peer that is reachable for one hop distance is also reachable for two hops distance, due to the network topology. In this case the peer will receive the same message, but with different *TTL*.

In this case the decision has to only taken into account the message with *TTL* = 1, and stored it at the peer by the "*storage_msg (msg)*" function. This decision has based on the network mobility. Since the peers are mobile, it's most reliable to store a message which stems directly from the source (*TTL* = 2), than a message which comes from a peer through (*TTL* = 1).

Because the node that send the other message (*TTL* = 1) may not be closest to the message source. After all the receive messages are stored, it's necessary to use this information to obtain all the possible combinations between the super peers. After the messages are stored, each peer will have in its memory, each neighbour view. Using all the received messages it's now possible to realize iterations over them using the function "*run_iteracions ()*". This function uses all the sources of those stored messages in order to have the local knowledge of the reachable peers at a maximum of 2 hops distance.

Using all the messages sources it's now possible to achieve iterations over them and to apply the chosen criteria's: Total n$^{O}$ of Reachable Nodes, Rating, Highest number of neighbour peers, Capacity, and finally the ad hoc module. Those criteria are in Figure 4-8.

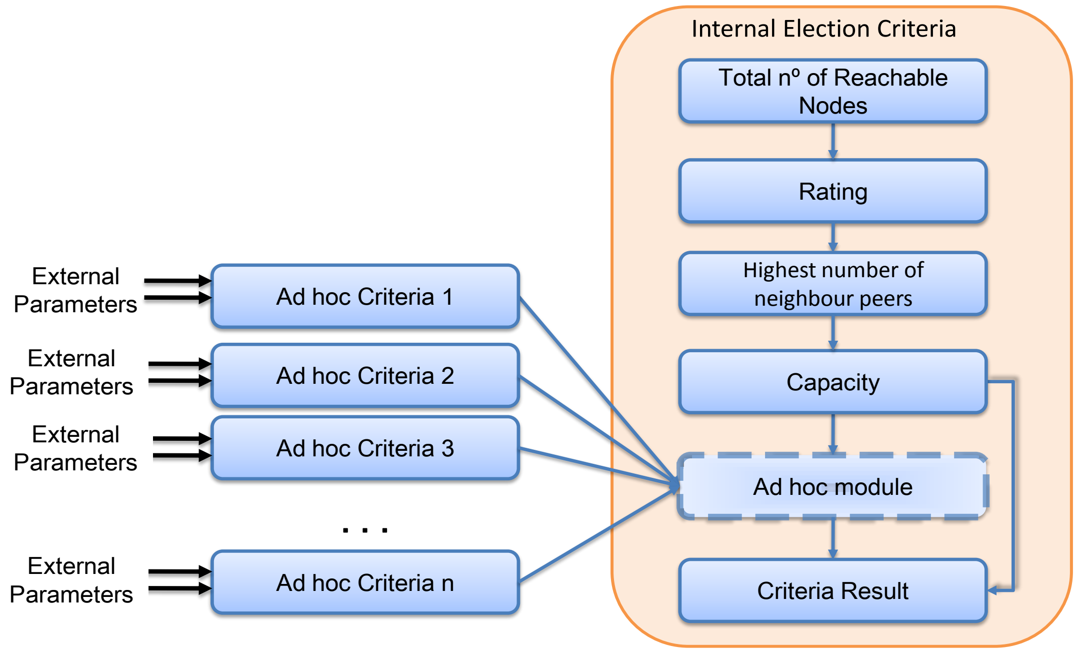


Figure 4-8 – Applied criteria used in the election Algorithm

After this, the algorithm has reached a solution and an election result message is send to all of the neighbours. This final election message is also sent with *TTL* = 2, since the peers at 2 hops distance were used for the election they also have to receive the election result.

If a peer receives a message with *TTL* = 2, it sends it back, and updates the *TTL* to the new value (*TTL* = 1). If the receives message have *TTL* = 1, then the peer stores the message and compares the contents of the message with the peer ID, and if they match, the peer passes to the super peer state. This process is in Figure 4-9.

Through all these steps used for the election process, the mechanism needs no central element to control or supervise. Each element in this network has its own uses and information that we use to “choice” wisely, using feathers of local view. These properties secured that the whole process is actually distributed, and as such has high flexibility and high scalability. The solution developed has a valuable feature, which should be further explored.

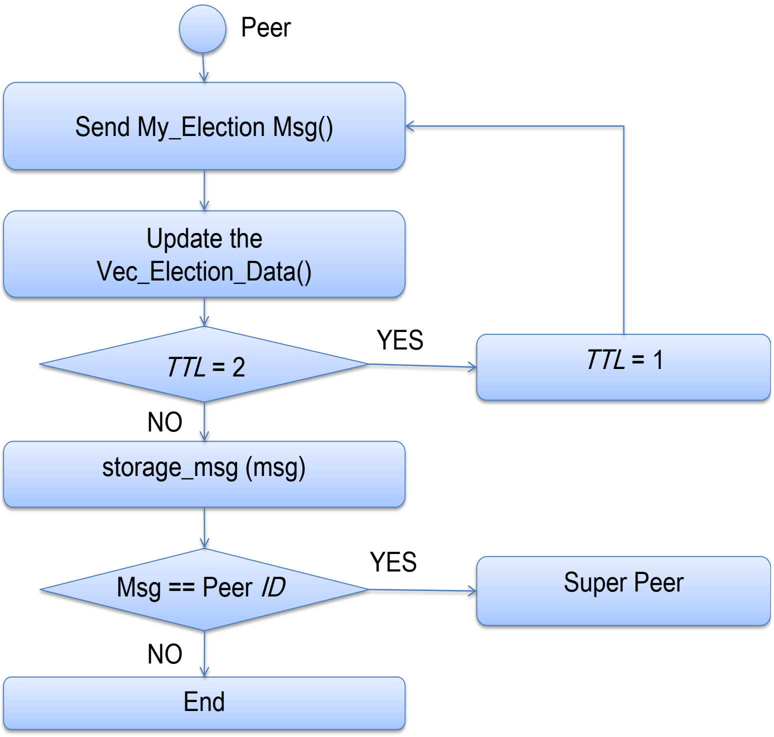


Figure 4-9 – Diagram of transmission of the election message

This feature is done by using an *ad hoc* criterion module, which uses input parameters. This means it’s possible to use other input parameters to conduct the election. Thus it is possible to adapt the process described in this dissertation in order to establish a result according to the objectives related to the need to for electing of a super peer.

```
class My_peer {
  Received neighbour msg () {
      Storage msg () ;
      Create the combinations ();
      Evaluate the reachable nodes for 1 hop ();
      Evaluate the reachable nodes for 2 hop ();
      Calculate each criteria ();
      Send election msg ();
  }

  Received election msg () {
      If ( msg content == node identifier) {
      peer.status = Super;
      }
  }
}
```

Figure 4-10 - Sample election algorithm code

Having described the framework specifications, it's now possible to generate e sample code for the algorithm. The sample code has all the function and procedures that the process has to perform, before election the super peer. The sample code to be implemented is described in Figure 4-10. This sample code has all the necessary elements that the algorithm has to have.

After all this steps are concluded, the election message is sent. The process of sending messages (Sequence Diagram) between the peer is described in Figure 4-11 - Sequence Diagram At this sequence diagram is visible that the algorithm will only exchange two messages between the peers, and indeed minimize the use of exchanged messages, that was set has the initial criteria. Because the entire method does not requires too much processor cycles in each peer, this method has all the conditions to run on any mobile or fixed device with the minimal processing power has a mobile phone, printer, laptop, etc..

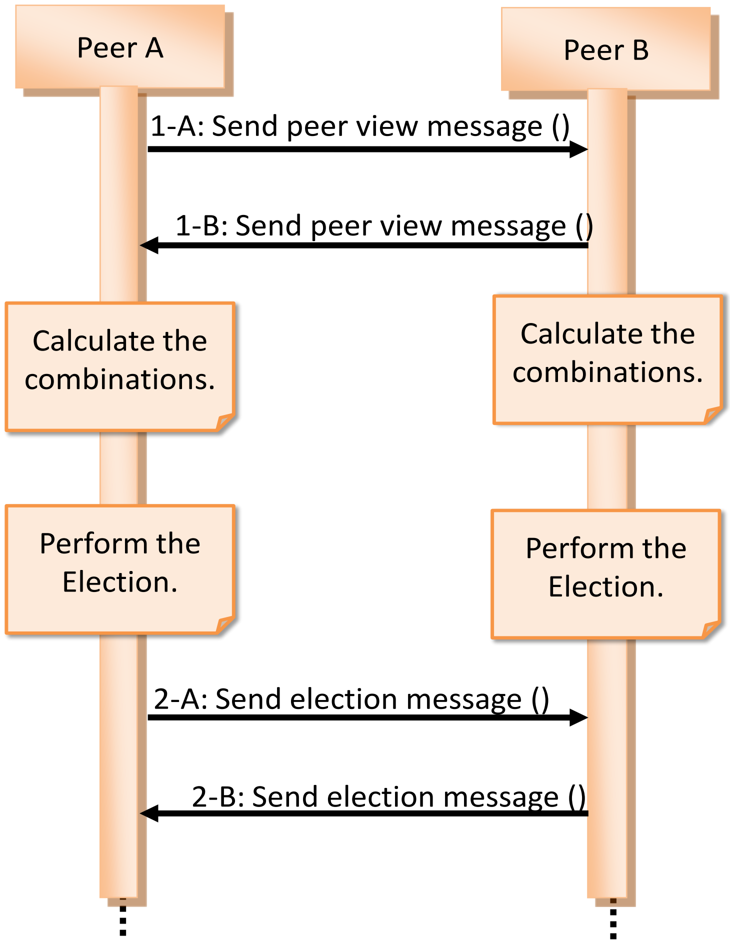


Figure 4-11 - Sequence Diagram

## 4.3 Summary

In this Chapter is described the approach selected for the election algorithm that aims to solve the problem of electing super peer´s within the use of a lightweight distributed super election algorithm. They also described the necessary steps in defining the algorithm, and at the same time, the approach chosen to solve the problems hindering the creation of the algorithm. The solution presented strengthened with the presentation of a specification supported by diagrams over particular decisions such as the internal election criteria. Sequence diagram of the exchanged messages are established, and is defined a "sample" code of the algorithm.

# CHAPTER 5 Testing and Validation

The main purpose of this section is to perform functional testing in order to assess the attainment of the system against the objectives that were set initially. Testing is the process of looking for bugs in the implementation of a system through experiments. Being that they are performed in a controlled environment where it simulates the actual use of the system.

The aim of testing is to prove a concept and verify its validity through the requirements defined and the available resources. Because systems can only be tested during a restricted period of time, testing cannot ensure complete correctness of an implementation or problems associated with the transformation from concept to implementation.

In order to validate the conceptualization made, it is necessary the definition of the used testing methodology and also the set of tests to be made on proof of concept implemented in accordance with it. The solution presented will be tested in several different scenarios, for which the methodology and application of the test should be identical. Listed below are both the test methodology adopted as the definition of what the tests and their results, ending with a final conclusion on the suitability of the solution set for assessing the compliments.

At the end of the chapter, validation tests will be made on the algorithm in order to assess its performance and efficiency. This chapter closes with the verdict on the suitability of tests and achieved results.

## 5.1 Methodology

There are several methods to test the suitability of solutions to meet their requirements, each with its specific field of application (Onofre, 2007). Although not all geared toward the same end, some have greater similarities the similarity being most evident is using the international standard for conformance testing of Open Systems as a starting point (Technology, 1991) . This is the ISO[5] IS-9646: "*OSI Conformance Testing Methodology and Framework*". As such, and since it necessary a testing methodology in the abstract, one will use the concepts defined by this standard.

[5] International Organization for Standardization

### 5.1.1 ISO/IEC 9646 – Framework and methodology for conformance testing

This standard was originally developed to provide a platform and define a terminology for the application of tests on OSI (“*Open System Interconnection*”) systems. But due to its low usage, the methodology has been little used for compliance testing of these systems. Nevertheless, the methodology has been applied to other types of protocols and systems being used as a basis for other methods of compliance tests, as used in standard *ISO* 10303 (i.e. “ISO 10303 part 30 – Conformance testing methodology and framework”).

The testing process described by this methodology is divided into three stages (evidenced in Figure 5-1). The first phase is the specification of an abstract test suite for the system in question, and is referred to as the definition of tests. The tests are abstract in that they are developed independently of any implementation. The second phase consists of defining the tests in order to be executed, and is called the test implementation. This stage is to take into account the implementation that will be tested, adapted to the tests defined prior to system implementation. The last phase, the testing, consists in its execution and observation of results. Which leads to a verdict on the compliance of the system under test with the initial requirements defined (Tretmans, 2001).

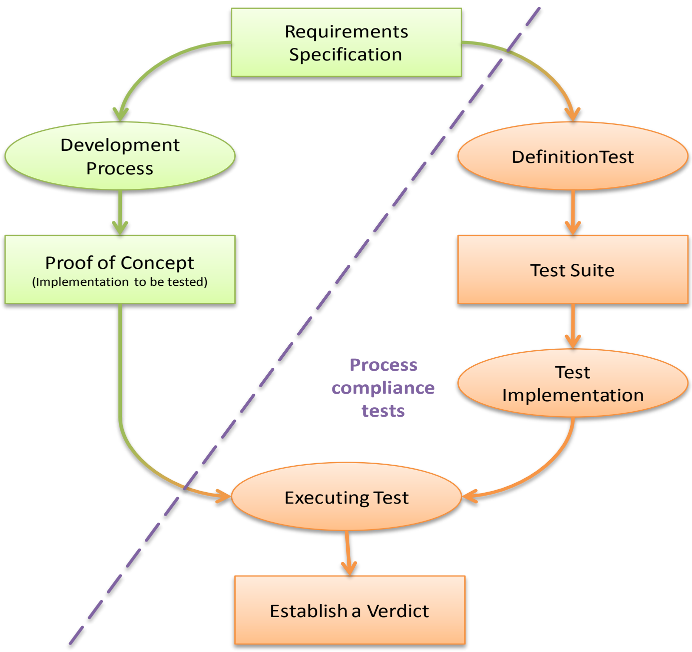


Figure 5-1 - Global view of the process of conformance testing

### 5.1.2 Notation for test – TTCN: Tree and Tabular Combined Notation

Because the testing methodology is standardized, it must be specified with a rating well defined, independent and globally accepted. The standard *ISO/IEC* 9646 recommends the use of semi-formal language *TTCN*, “Tree and Tabular Combined Notation”.

This notation is governed by the black box model meaning that the internal behaviour of the system is not relevant. For the black box model the system is treated like a black box, whose functionality is determined by observing it and no reference is made to the internal structure of the program. The used notation also appears on the tabular form the various parts that define the test, these being the overview about the test, the necessary declarations for implementation, restrictions and dynamic part. This latter, in the dynamic part, the tests themselves are described, namely its behaviour (Tretmans, 1992).

In *TTCN* the behaviour of the tests is defined by the sequence of events that occur during the test. A sequence may have several alternatives in which different behaviours can be considered, depending on various system responses to requests made. This behaviour is defined in a tree form, and a chain of successive events is indicated by increasing the indentation of several events, and the alternative events are defined using the same indentation.

Since the sequence of events ends with the definition of a verdict, so that different behaviours may take different verdicts, for instance there may be more than a behaviour that matches the success of the test.

### 5.1.3 Proof of Concept

In order to be able to perform tests to validate the concept of the proposed solution that meet the challenges encountered, it is necessary to the creation of an implementation that can be brought into a process of validation through experimentation. This implementation will be not a complete implementation but only a functional implementation to assess the validation of concept.

This implementation will be part of the third level (Proof of concept demonstrated analytically and/or experimentally) the Technology Readiness Levels. It stipulates that this step includes both the analytical study for the definition of appropriate technologies to be used as tests in a controlled environment in order to validate the concept of the solution.

These studies and tests are presented without aim to be a functional implementation with all the capabilities inherent in a commercial product. It is therefore necessary to consider the tests in order to realize the potential of the proposed solution and not the specific performance of the implementation carried out. Therefore it is necessary to point out the faults and improvements to get these documented, which in the case of a commercial implementation would help address some root problems (Mankins, 1995).

After the tests are defined, then a survey is made in the implementation of the technologies used in the simulation, the solution, and the presentation of how they were applied to carry out the proof of concept. The tests are studied in a controlled environment in 5.4 Test Execution.

Therefore the aim of testing is to gain confidence that if the system is used in an uncontrolled environment it will work satisfactorily. Tests can only prove the presence of errors, but a successful test does not imply that the system is error-free (Tretmans, 2001).

## 5.2 Test Definition

Defined the methodology to be used in the testing can proceed with the definition of use tests that will result in the verdict on the suitability of the solution found in compliance. Having regarded the compliance for the requirements, there will be undertaken two major tests. The first test will occur in networks with a very small number of peers, in order to evaluate the result. Also in this first test the topology will be pre-defined. The first test starts with the establishment of the network topology. It then passes through the election mechanism, and finally the super peers are elected amongst all the peers present in the created topology. It could be used other approach, but this proved to be the faster and more intuitive. The use of visual analysis has the method to confirming the success or failure of the test is due to the fact that a peer network has no global knowledge of all network peers, and only a visual analysis can measure the success or failure over the test.

After analysing the test results in different small size networks, there will be defined a new test for the network where the topology is totally random and the number of peers will be higher compared to the earlier test. In this second test, it be evaluated the election over random topologies, and on a large-size network, to verify the scalability of the algorithm.

Defined the methodology to be used to the achievement of the tests, then it proceeds to the definition of the use tests. This will result in the verdict over the suitability of the solution found in compliance. In this case the tests will be performed manually, mainly because after running the algorithm, only a visual analysis can validate the result. Below there's a synopsis of the tests defined.

### 5.2.1 Election Test – Fixed Topology Election

The intent of this test is to validate the election result on a pre-established network topology. First the topology is defined and validated then the election occurs and the reachable peers are evaluated. The topology of the network created must be the desired before the test takes place. It is now presented a summary of the implementation.

| **Test** | |
|---|---|
| **Test Name:** | Fixed Topology Election |
| **Group:** | Fixed Topology |
| **Purpose:** | Validate the Election |
| **Comments:** | This test is conducted to validate the election. Is applied to a pre-established fixed topology and them the validation of the Election |

| **Behaviour** | **Verdict** |
|---|---|
| ! Set the desired topology | |
| ! Verify that the resulting topology was the desired | |
| ? Topology successfully created | |
| ? Election occurred in all peers | |
| ! Check peers elected | |
| ? All peers are reachable | Success |
| ? Not all peers are reachable | Failure |
| ? Election not occurred | Failure |
| ? Topology corrupted | Failure |

Table 5-1 - Details on the Fixed Topology Election Test

This is used to confirm the creation of the topology, to ensure that all peers are actively involved in the election mechanism and that by the end of the test is possible to verify that the election occurs.

After the creating of the topology it´s necessary to ensure that all the peers in the network are in fact actively involved in the election mechanism. If this does not happen, the test fails. In the final phase of the testing it´s necessary to determine whether the election in fact, enables each peer in the network to make use of the distributed database in this super peers to reach all other peers in the network. If all these points are checked the test achieved success.

### 5.2.2 Election Test – Random Topology Election

Those simple test cases to evaluate the topology and the algorithm behavior are of very importance, since the tested cases represent several specific topologies present in the network. Although the algorithm needs to provide also an efficient election regardless, the network structure but also taking into accounts it scalability.

Having the results for specific networks the next test evaluates the performance over a large random topology.

<table>
<tr><th colspan="3">Test</th></tr>
<tr><td>Test Name:</td><td colspan="2">Random Topology Election</td></tr>
<tr><td>Group:</td><td colspan="2">Random Topology</td></tr>
<tr><td>Purpose:</td><td colspan="2">Validate the Election</td></tr>
<tr><td>Comments:</td><td colspan="2">This test is conducted to validate the election. Is applied to a pre-established Random topology and then the validation of the Election</td></tr>
<tr><th colspan="2">Behaviour</th><th>Verdict</th></tr>
<tr><td colspan="2">! Set the desired topology</td><td></td></tr>
<tr><td colspan="2">    ! Verify that the resulting topology was the desired</td><td></td></tr>
<tr><td colspan="2">        ? Topology successfully created</td><td></td></tr>
<tr><td colspan="2">            ? Election occurred in all peers</td><td></td></tr>
<tr><td colspan="2">                ! Check peers elected</td><td></td></tr>
<tr><td colspan="2">                    ? All peers are reachable</td><td>Success</td></tr>
<tr><td colspan="2">                    ? Not all peers are reachable</td><td>Failure</td></tr>
<tr><td colspan="2">            ? Election not occurred</td><td>Failure</td></tr>
<tr><td colspan="2">        ? Topology corrupted</td><td>Failure</td></tr>
</table>

Table 5-2 - Details on the Random Topology Election Test

For this test its validation is subject to the same requirements as above, but in case of a random topology. Besides being necessary to verify the created topology, it is also necessary to confirm that all peers participating in the mechanism are also reachable. If all these points are checked the test achieved success.

## 5.3 Implementation

After defining the tests, these will be applied to the solution created. The creation of the solution was to define structures to record the messages and create tasks associated with the creation of the necessary criteria. To check the result of the test, it must be used a simulator that represents identically the environment where the algorithm should be applied.

The simulator which suits better the needs is the *JXTA-Sim* (Meier, 2010) that satisfies all the initial requirements regarding the Scalability, Extensibility, Usability and also Flexibility. The simulator provides support for overlay networks, visualizations and the ability to gather statics, accurate results, has good support and documentation and the ability to interface with the Java language.

For getting the election algorithm results, the simulator offers different types of methods to evaluate the algorithm performance regarding tables, vectors, structures or any other local variable, it's possible to record information on the number of exchanged messages and other statistics options and to record all in a file. Documentation was another factor in choosing the framework for the election. In this simulator the performed experiments have allowed to test up to 10.000 peers. The *JXTA-Sim* has been implemented following an object oriented approach which easily allows extending the simulator in order to evaluate other aspects of *JXT* [6]. Also the *JXTA-Sim* design tries to be simple and the code is well commented to make it easier to understand. Due to the way it was implemented it allows evaluating the *JXTA* lookup algorithm by performing different experiments with different parameters. Users can obtain different types of results such as graphs and statistics of the simulated scenarios and they can also code new tests to obtain other results. The *JXTA-Sim* also provides accurate results, have good support and interface with java, for all of the *JXTA-Sim* was the chosen simulator.

[6] A set of open protocols that enable any connected device on the network, ranging from cell phones and wireless PDAs to PCs and servers, to communicate and collaborate in a P2P manner.

To develop the election algorithm, there was only a technical requirement that specify that the language used would be Java. This language is platform independent, supports inner classes. In addition is a free platform, free of charge that has extensive libraries of routines that facilitate the cooperation with protocols. To develop the application environment chosen has the Eclipse. As this field there was no restriction, the choice was due to it has been already used before and already possess knowledge in this field.

The next stage was to create the election algorithm in each peer using the implemented classes but also implementing new ones. To evaluate the efficiency of the algorithm, all the evaluated tests were taken in a full controlled environment by the *JXTA-Sim* simulator. The simulator has already implemented the *JXTA* protocol that was extended to the election algorithm.

Through the *JXTA* protocol, the main functions were already developed and ready for use, including the entire structure of super peers and edge peers, and also the indexing service used by the super peers. For the simulation, there will be no peers leaving the network, or failing, the index and the queries will not be addressed, since this is not the focus of this dissertation. The purpose of the simulator is to validate the election using the selected tests. The result of the election in each peer is examined through a file created for this purpose, provided by the simulator methods.

The *JXTA-SIM* will provides information over the status of a peer in the overlay,  to verify that the election actually occurred, and the number of messages exchanged by the peers to to check the frequency that messages are exchanged and their number. For the tests were created two files, one containing the identifier of each peer and another with the connections between the necessary peers, to define the network topology. The validation of the network topology will be made with the help of the tool *yEd Graph Editor*[7], used for viewing the topology created through the simulator, and to provide a visual analysis.

[7] Application that allows to aplly automatic layouts to a range of different diagrams and networks.

## 5.4 Test Execution

After defining the previous tests, they will now be applied to the distributed super peer election algorithm.

### 5.4.1 Fixed Topology Test

This type of analysis serves to verify the results of the developed solution under a known topology. This is mainly because the implementation of the solution should be efficient but may also take into account real situations. There is a set of known topologies that can be found in different day-to-day contexts that difficulties the setting of a semi-centralized architecture. The use of the fixed topologies present in real life will help bring closer the applicability of the solution with the today infrastructure.

#### 5.4.1.1 Ring Topology

The first topology used for the fixed topology test is the “Ring” (Figure 5-2 - a) topology. The choice of this topology is due to the fact that the election mechanism in these conditions is ambiguous, given the variety of possible solutions in this topology, the application of the solution. Since the topology is a ring, there are several possible and acceptable solution for the establishment of the super peers infrastructure. After creating the topology and the election occurred in all the peers, with the election result the peers are in fact capable of reach all the others peers. This test leads to the election of all the peers in the network and since the aim of the election was reached, it is concluded that the test was a success. The result of the test is visible in Figure 5-2 – b).

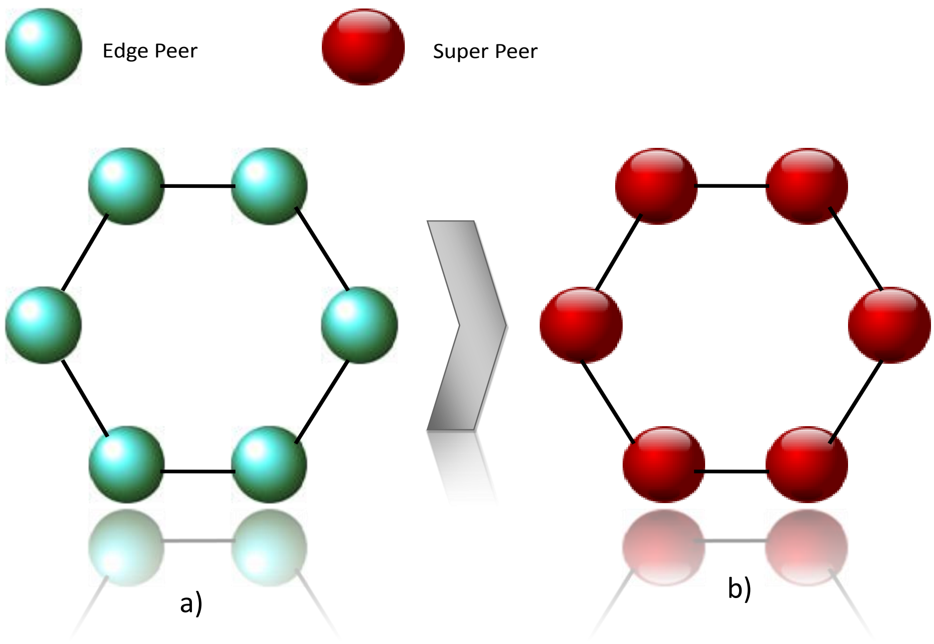


Figure 5-2 - a) Ring Topology before the election, b) Ring Topology After the election

### 5.4.1.2 Mesh Topology

For the second topology “Mesh” (Figure 5-3 - a), the establishment of semi-centralized structure is no longer so vast, because the existence of two nodes with a high number of links relating to all others. After the topology has created and the election occurs the elected peers are consulted.

The election that resulting from this test provides each peer with the possibility to reach all the others peers, resulting in the final election of two peers. The test behaviour ends with success, since it’s visible in the Figure 5-3 – b) that each peer present in the network is capable of reach all the others peers through the distributed super peers’ index.

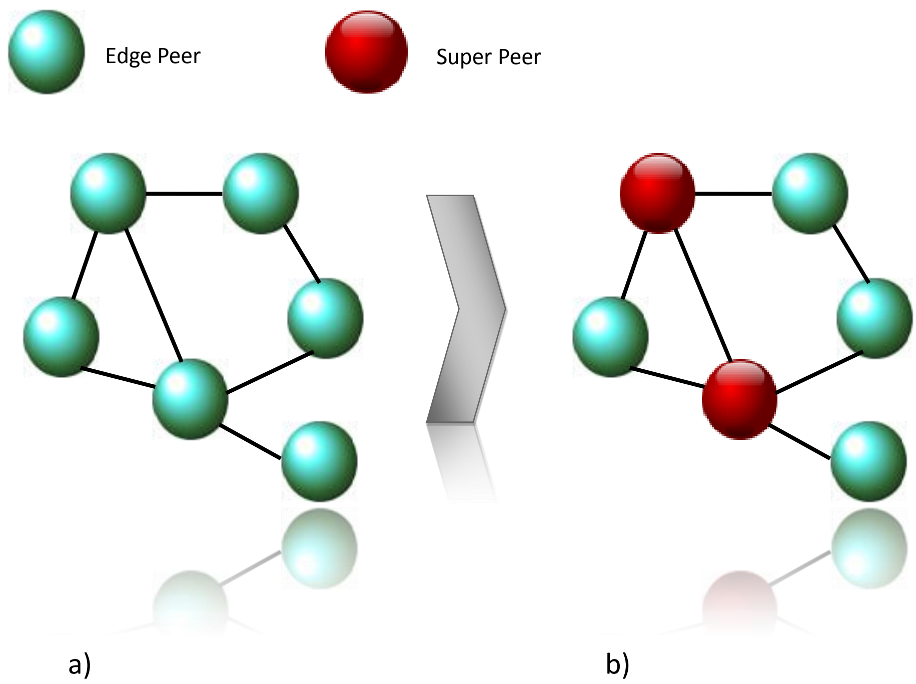


Figure 5-3 – a) Mesh Topology before the election, b) Mesh Topology after the election

### 5.4.1.3 Star Topology

For this third test is used the “Star” topology, in Figure 5-4– a). The use of star topology is mainly due primarily to confront the solution developed with a topology in which there is a clear solution. The most obvious solution to an election by a network administrator will be the election of a single central node, which should serve to indexing mechanism for all others. The desired topology has achieved. This topology also passes the second phase (where the election occurs in all the peers), and the third phase where all peers are reachable.

Since the election results in a topology were two peers are elected super peers, and with only those two peers, it’s possible for all nodes to reach all the peers present in the current topology, so the test was considered a success. The final result from the election is visible in Figure 5-4– b). Although the final solution is not just electing a super peer for the election of only two super peers is regarded as a near-optimal solution under these conditions.

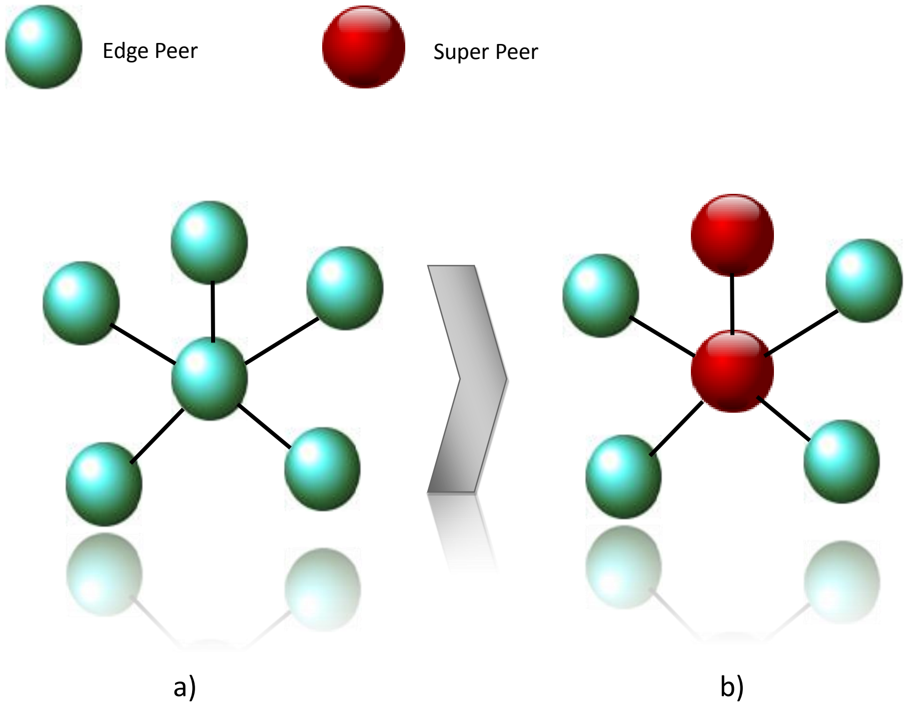


Figure 5-4 – a) Star Topology before the election, b) Star Topology after the election

### 5.4.1.4 Tree topology

Other tested topology is the one visualized in Figure 5-5 – a), the “Tree” topology. The test applied to topology, result in the success of the first and second phase, were the peers and the connections established are the desired ones, and all the peers present in the network reach a result in the election algorithm. In this topology tree peers are elected super peer. After the previous phases, the last phase established the success of this test, since all the peers are reachable with the elected super peers (Figure 5-5 – b).

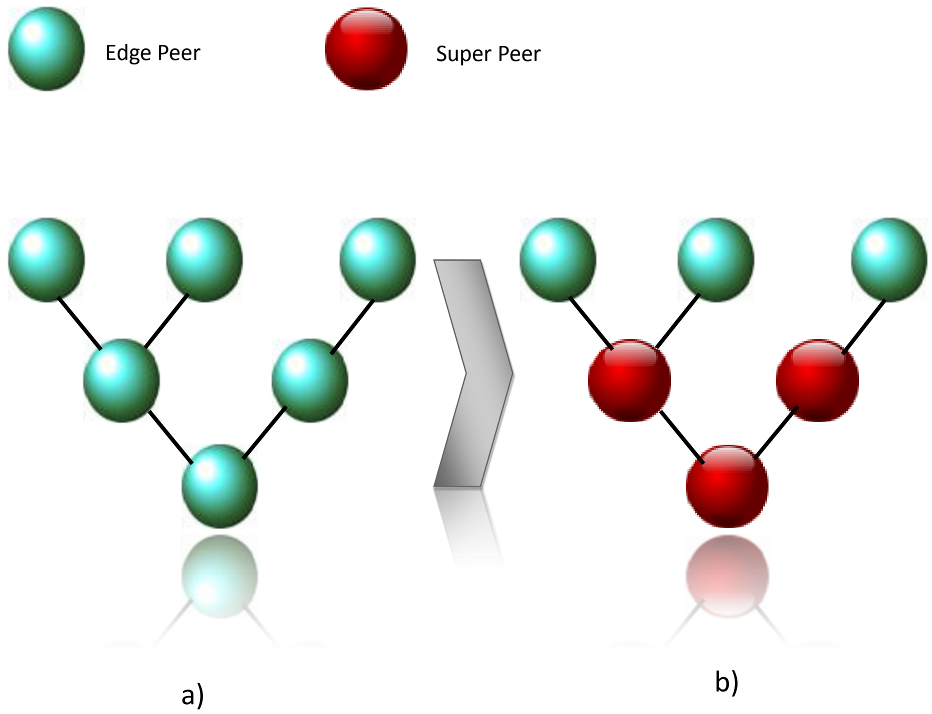


Figure 5-5 – a) Tree Topology before the election, b) Tree Topology after the election

### 5.4.1.5 Bus Topology

Another used topology is the “Bus” topology, with this topology the election test provides not only the fixed desired topology, but also that the election takes place on each peer. Also the total number of peers in the system is possible to be reached with the two super peers elected. The most obvious solution to this election by the network administrator will be the election of a single node, which should serve to indexing mechanism for all others, since all of them are connected to each other’s. With this test is achieved the verdict of success.

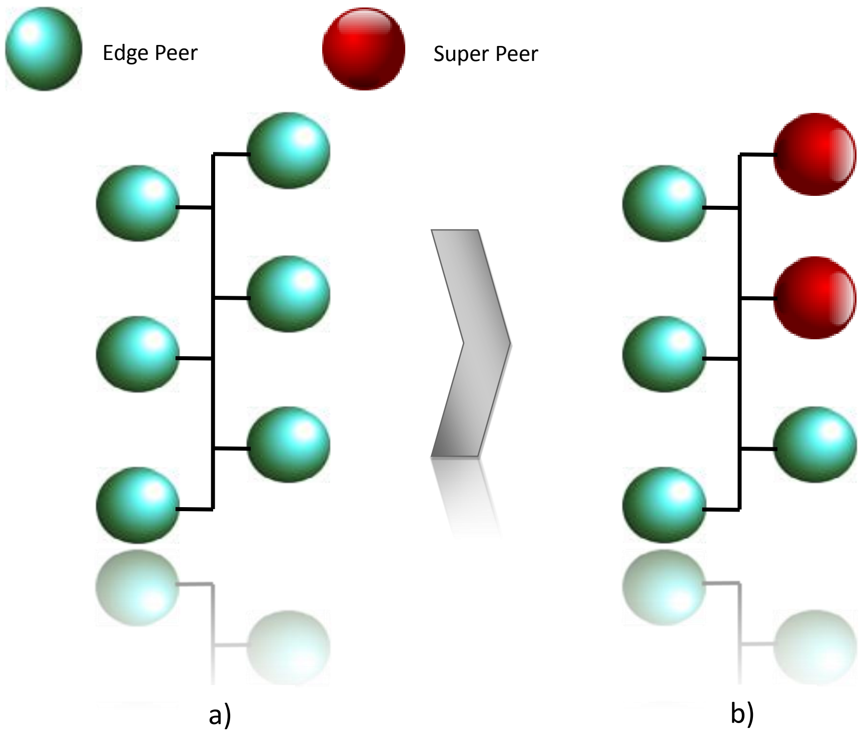


Figure 5-6 – a) Bus Topology before the election, b) Bus Topology after the election

#### 5.4.1.6 Fully Connected Topology

Using the "Fully Connected Topology" (Figure 5-6- a)) the simulator provides the outcome of the election algorithm that is visible in Figure 5-6- b). This test applied to the Fully Connected Topology, result in passing of the first and second phase, were the peers and the connections established are the desired ones, and all the peers present in the network reach a result in the election algorithm. After passing in the previous phases, the two elected super peers provide that all the peers are reachable for this topology and established the success of this test. In this topology where all peers have links between them, occurs the election of two super peers, in a set of six possible peers.

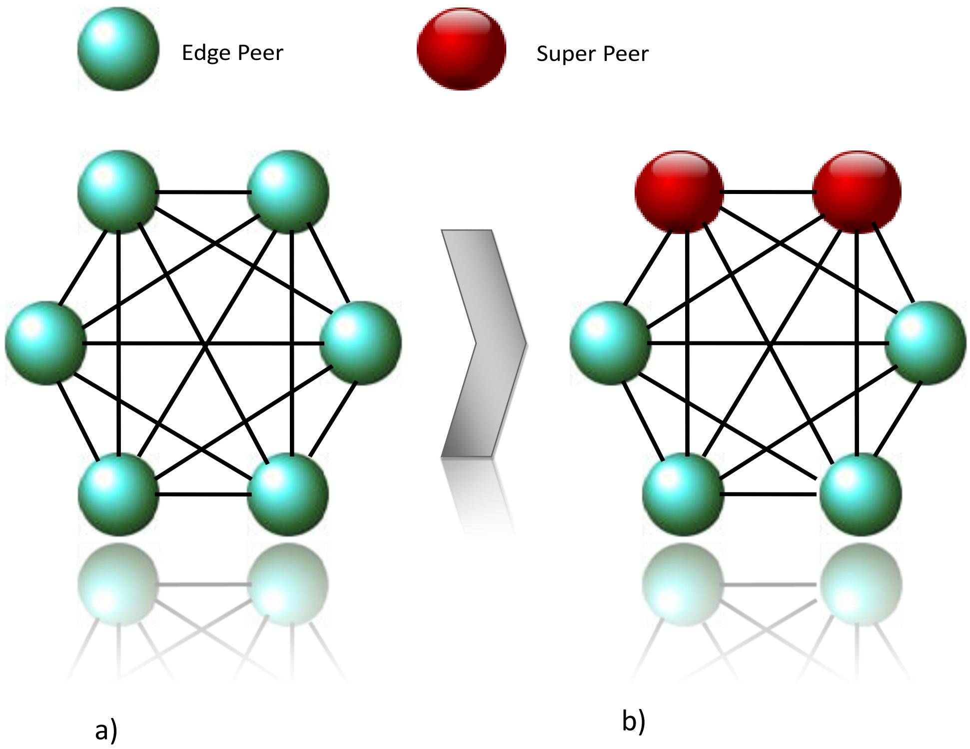


Figure 5-7 – a) Fully Connected Topology before the election, b) Fully Connected Topology after the election

### 5.4.1.7 Line Topology

The fixed topology test, applied to the "Line Topology" Figure 5-8 – a), result in the peers and the connections established are the desired ones, and all the peers present in the network reach a result in the election algorithm. After the previous phases, the last phase also seen in peers elected for this topology and thus established the success of this test.

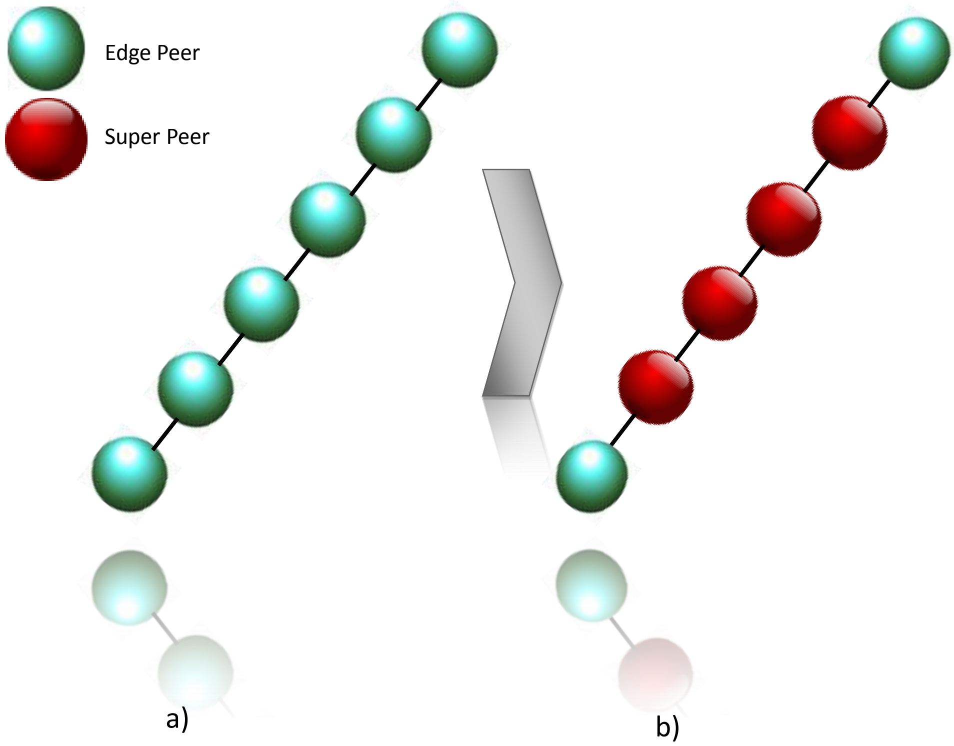


Figure 5-8 – a) Line Topology before the election, b) Line Topology after the election

### 5.4.1.8 Square-Edged Topology

Other tested topology is the one visualized in Figure 5-9– a, the “Square-Edged” topology. The test applied to topology, result in the passing of the first and second phase, were the peers and the connections established are the desired ones, and all the peers present in the network reach a result in the election algorithm. In this topology five peers are elected super peer. After the previous phases, the last phase established the success of this test, since all the peers are reachable with the elected super peers Figure 5-9 – b).

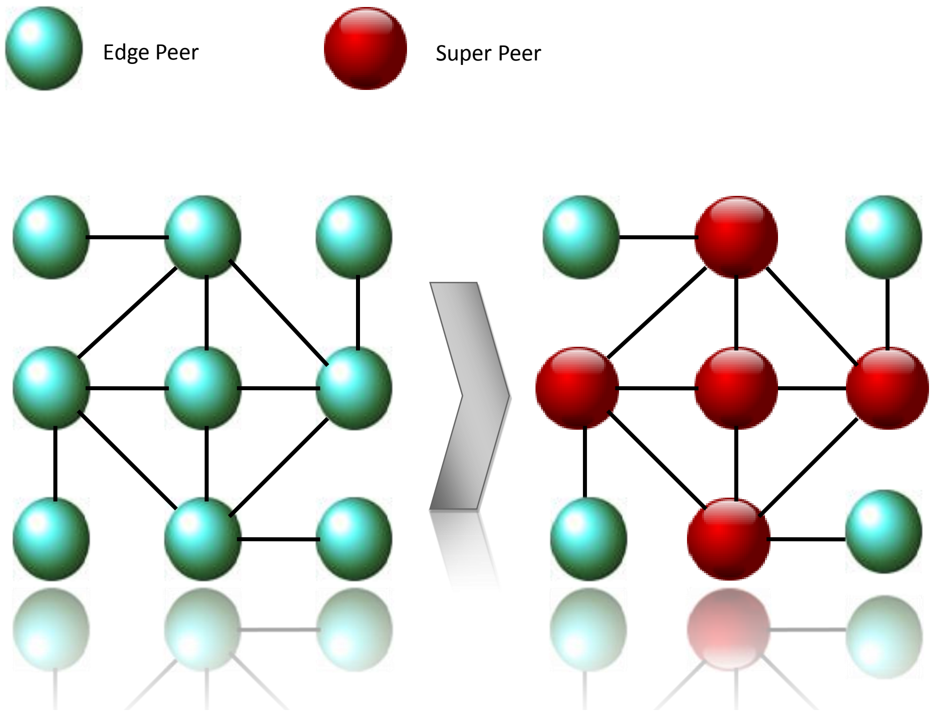


Figure 5-9 – a) Square-Edged Topology before the election, b) Square-Edged Topology after the election

## 5.5 Random Topology Test

After being displayed various topologies over the first test, now it will be taken in the outcome of the second test. The use of a random topology should also be taken into account. Because peers are capable of mobility, the structures above can be changed and modified the use of a random topology reflects this situation. Associated with a random topology, it was also decided to use a high number of peer to evidence the behaviour of the algorithm over large-scale network.

### 5.5.1 Large-scale network

For this particular case, was verified that despite the choice of leads and peers in the network is random (it was created without imposing any restrictions or specifications), by the end of the first phase the result of the topology was in line with the established topology. In addition to the topology, this test also uses an excess of peers over the network. The choice of the number of peers fell to the number 59, but could have been any other value.

For this case, the simulator uses a fixed number of 59 peers. The displayed result is not presented here due to space issues. After verifying the topology is noted that all 59 peers in the network, actively participated in the election mechanism, and all obtained a result. With visual analyses is possible to establish that in fact all the network peers are reachable. Therefore, it can be concluded that the verdict of this test was successful.

## 5.6 Verdict and Summary

The first conclusion to be drawn from the tests is that the proof of concept successfully managed to elect the super peers, through the active participation of all peers in the network, thus demonstrating that the algorithm managed to fulfil its function.

Although in some cases analysed the election results weren´t the ideal, the final result of the algorithm presents a very approximate optimal solution. In certain topologies there was the election of a small number of additional super peers, which in fact offer no advantage. These additional super peers can be considered as being in excess, but as the end result is quite satisfactory and is consistent with the proposed objectives, the end result may be regarded as very satisfactory and efficient. After the tests it was found that the number of messages exchanged between peers is greatly reduced. This exchange messages are from the ones sent by each peer, and also for the messages returned by them.

The fact that provides greater efficiency to the election mechanism would not be much advantage over these results. Because this facility is designed to be used in wireless networks, and the use of small number of messages exchanged are considered a priority. As the final results give the network an already low number of additional elected super peers and also a lower number of messages exchanged, the algorithm is considered to have an efficient performance for both small networks and for large-scale networks.

# CHAPTER 6 Conclusions and Future Work

This dissertation focus on the Super Peer Election, that involves the selection of a subset of the peers to serve a special role (Super Peer). Was necessary to raise the key concepts (*P2P, MANETs,* etc.), to provide a better understanding of the questions and to frame the problem and to examine their limitations. Were made a review over the existing literature that provides the background for the establishment of some current Super Peer Election Mechanisms.

It has then examined the different approaches that could be taken to resolve the problem and the challenges that the election mechanism has to take into account. The challenges are driven from the nature of the network, and are related to the environment (*P2P* systems over a *MANET* structure) where the algorithm is intended to be applied. There were a survey over the existing election mechanisms and by the final were discussed the relation between the challenges and the existing elections algorithms. Using this analysis as concluded that existing implementations do not present a concrete solution to the problem. Although some approaches discuss the criteria and establish some requirements for the establishment of an efficient and effective election, in most cases, their use is limited by parameters, not giving a faithful representation of the needs imposed for a distributed election.

Was then presented the definition and the design of the election mechanism, in order to respond to the Super Peer Election Problem according to the set of predefined super peer election criteria. After an extensive analysis of the existing approaches and a survey of the initial problem involving the election of super nodes in a distributed network, with heterogeneous peers', was implemented an algorithm strengthened by theoretical foundations that intend to endow a solution to the election problem.

This original implementation, involves the selection of a subset of the peers in a large scale network, and the further election of super nodes in a dynamic network to serve a distinguished role. In order to validate the concept, it was then defined the used testing methodology and the battery of tests to be made on the proof of concept. In all these test cases the existence of one or more node with a higher number of connections to others who are their neighbours, it implies that they will always be elected as the Super Peer.

After executing the tests for fixed, random, small and large-scale networks, the election method proved to be a good approach for networks which have an imbalance in terms of connections that each node has. It was thus proven that it is possible to perform the algorithm and conduct a distributed election that takes into account all the theoretical requirements established initially, and that can be applied in any situation not being limited by parameters/variables or specific uses that the network can present.

## 6.1 Future work

The algorithm can be optimized through a careful management of the messages so that the outcome of the election result is identical to the "ideal" one.

Can be added more criteria for the election algorithm using different ad-hoc modules. These *ad-hoc* module should contain criteria that might be adapted to the context to which the network is being used, so that the election is affected externally. The development approach not only provides the mechanism with the ability that is not be restricted to constraints imposed by the intrinsic characteristics of the system, but also possess the ability to adapt to needs not previously considered. This adaptation allows the development of future new eligibility criteria, which will allow the continued use of the algorithm regardless any future restrictions imposed by the network.

## 6.2 Publications

This development has made within the research activities of the *CTS* (Centre of Technology and Systems) in the *GRIS* (Group of Research on Interoperability System), at Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa. Supported by the European research and development project: FP7-216 420 CuteLoop "*Customer in the Loop: Using Networked Devices enabled Proactive Intelligence for Customers Integration Drivers of the Integrated Enterprise*", between September 2009 and September 2010.

Alongside the writing of this dissertation, there was also contributing to the following literature. The lifting of the Super Peer Study of Election Methods, directly contributed to this deliverable: "*D1.1 - State-of-the-Art Analysis".* The concept defined in the dissertation was an integral part of these two deliverables, as one of the research questions of the Project: "*D1.3 - CuteLoop Concept, D1.4 - CuteLoop Concept - Public Report*".

The specification of the dissertation entered directly in this document: “*D2.1.1 - Specification of Basic Services”*. The implementation was used directly in the project, and part of the final prototype: “*D2.3 - Basic Services Full Prototype*”. The tests carried out in the dissertation, they go directly into this report, at the distributed election section: “*D6.2 - CuteLoop Full Prototype Testing and Assessment Report*”.

# ANNEX A SIMULATORS OVERVIEW

This section surveys existing *peer-to-peer* simulators and theirs features. Are described the various network simulators, their features, and their utility in a short *P2P* simulators overview. Are also specified features that are only supported by a restricted set of simulators.

## I. Anthill

Anthill was developed by the department of Computer Science of the University of Bologna and was presented in 2001 (Ozalp Babaoglu, 2001). This simulator is an agent-based *P2P* system simulator that attempts to use certain aspects of biological and social systems used for *P2P* application development, deployment and testing. This social aspects driven form the behaviours of interconnected nest of ants, in order to use the nest as a representation of the peer in the network, and an ant represents the exchanged messages between the peers (nest), as each ant moves from nest to nest, also the messages are exchanged from peer to peer. Anthill is written in Java and based on the *multi-agent system* (MAS) paradigm. MAS systems are collections of autonomous agents that can observe their environment and perform simple local computations that leads to actions based on these observations. The agent may have non-deterministic behaviour, which leads to the modification of the environment as well as the agent´s location within the environment.

For this simulator each nest is capable of performing computations and host resources, but it also provides facilities such as storage management, communication and topology management. It is also provided a simulation environment to help developers to analyse and evaluate the behaviour of the *P2P* systems. To select the simulation parameters it is used an XML, that parameterizes the structure of the network, the ant algorithms to be deployed, the characteristics of the workload presented to the system, and also the properties to be measured. Anthill is designed to be easily deployable for either simulation or real world use. This is accomplished by the use of the JXTA (*Juxtapose*) project promoted by *Sun Microsystems*. With the use of JXTA provides the possibility of using many different transport layers for communication, including TPC/IP and HTTP, but also deals with the problems related with firewalls and NAT.

The use of a JXTA service used to design the runtime environment provides the exploration of all the facilities offered by the JXTA core that provide the infrastructure for the construction of ant-based P2P distributed applications. The simulation environment enables programmers to evaluate several different experiments and obtain average figures for statistics, and for monitoring the network traffic the only approach is at the ant level since at the packet level this feature isn´t available. An application called Gnutant has developed for document sharing and shares some characteristics with the Gnutella application, but Gnutant inherits the free search capacity of Gnutella, without relying on inefficient broadcasting techniques. Despite its many promising features, development on Anthill has been stopped in favour of a more scalable and light-weight simulator called PeerSim.

## II. PeerSim

PeerSim is a *peer-to-peer* simulator developed under the BISON (Biology-Inspired techniques for Self Organization in dynamic Networks) project and is available on Source Forge (*http://peersim.sf.net*). PeerSim is the successor of the Anthill framework and has been developed by the same authors. The stated goals of PeerSim are extreme scalability, support for dynamicity and modularity of simulator components. Jelasity et al. propose several simplifying assumptions to support extreme scalability in PeerSim. Low-level or packet-level details of the underlying network are not simulated, and neither the bandwidth or latency.

A description over the simulator is presented in (Alberto Montresor, 2009). This simulator is developed to be designed with modularity in mind and ease of configuration. It offers the possibility to add new nodes or destroy existing ones or even modify their parameters. The simulator is presented has having cycle-based and event-based simulator engines and both can be fully configured and customized. The cycle-based engine allows the simulator to scale up to a larger number of nodes (one million nodes), but this does mean some accuracy is lost, in comparison with event-based engine. Each simulation is specified by a plain text configuration file, very similar to a Java property file, and since its implementation is over the Java language, it allows building experiments at run-time via reading configuration files and dynamically loading classes. With this simulator the using the cycle-based engine, the limit of the network size is practically the entire available memory, and networks of more than $10^7$ nodes have been simulated. For the event-based engine and with the use of most complex protocols, the network still scales up to $10^5$ nodes or more.

Using the configuration manager is possible to load the configuration files or the command line parameters that decide what form the simulation will take. The configuration manager also provides the user to specify which other components that should be loaded, giving the developer the possibility to create their own interchangeable components to be used during a simulation allowing pluggable components.

According to the authors, PeerSim is capable of performing simulations on networks of up to 1,000,000 nodes. Over the PeerSim development site the implementation was developed in Java and was registered the existence of predefined protocols for this simulator, namely OverSat, SG-1 and T-Man and others example protocols named Aggregation and Newscast. PeerSim developers have released documentation and mailing list for support, but the mailing lists are not used for support, and provide release announcements and changed logs. The others available documents are released in the form of Javadocs, tutorials and by protocol implementation guide. Another register feature is that the simulator does not provide visualizations over the simulation. The Project is now partially supported by the Napa-Wine (*Network-Aware P2P-TV Application over Wise Networks*) project.

## III. P2Psim

P2Psim is a free, multi-threaded, discrete event simulator that can simulate structured overlay only, and has developed by Gil et al. of the Computer Science and AI Laboratory at the Massachusetts Institute of Technology. P2Psim intent to evaluate, investigate, and explore *P2P* protocols and as part of the IRIS project. It´s main goals are to make understanding over *P2P* protocol source code easy, to make comparing different protocols convenient, and to have reasonable performance. P2Psim adopted the use of threads in simulations, and Gil et al. defend this approach stating that the use of threads makes “the implementations (of *peer-to-peer* protocols) look like algorithm pseudo-code, which makes them easy to comprehend”. The use of threads is accompanied by the use of a event queue to store pending events sorted by a defined time stamp, that shorts the event´s but also removes them from the queue and execute the first one within a the new thread. This provides the main thread to stay blocked, until all new threads have finished executing. For this simulator there are register many peer-to-peer protocols, examples of those protocols are Chord, Accordin, Koorde, Kelips, Tapestry and Kademlia and P2Psim is written in C++.

Those protocols are based on the concepts of mapping node identifiers to internet protocol addresses, and so no actual applications for fire sharing have been implemented. This simulator has established has principal components the Node, the Network, the Topology, the Lookup Generator and the Churn Generator objects on with the nodes represent individual peers which belong to the Network. The Topology object deals with the latency of communication between any two peers in the Network and the Churn Generator is responsible for model the dynamic aspects of the Network, such as the arrival and the departure of Nodes.

In terms of the current state of the simulator a multi-tier capability is not known, it does not provide any simulation visualisation or a GUI approach. But by the other side it can be used in conjunction with a third party GTK application to provide a GUI, and perl scripts are provides for the generation of graphs. It scalability is only registered over 3000 nodes, and only using the Chord protocol, although the authors believe that it should scale well to approximately 10,000 peers. However it has very little documentation which makes it difficult to extend.

## IV. PlanetSim

PlanetSim is an object oriented simulation framework for overlay networks and services. With PlanetSim, the developers can work at two main levels: creating and testing new overlay algorithms like Chord or Pastry, or creating and testing new services (DHT, CAST, DOLR, etc) on top of existing overlays. Besides it doesn´t include a trivial P2P simulation sample simulation, Chord and Symphony simulations exist and simulations are expected to scale to 100,000 nodes. PlanetSim is developed in the Java language it's a very basic simulator and with a fast learning curve but it has no mechanism to gather statistics. The simulator is set with a layered architecture, where the different elements can be replaced easily to adapt the simulator correspondingly, and to provide flexibility and extensibility (Jordi Pujol Ahulló, 2007).

PlanetSim supports static visualisations with GML or Pajek outputs of network topologies, the simulation can be saved to disk for reuse, and Multi-tier simulations are available as third party components and are achievable with this simulator. In the documentation is referenced that the Gnutella *P2P* protocol will be implemented. PlanetSim does seem to have good variety of support, provided as Javadocs, tutorials, presentations, developer documentation and mailing lists.

## V. Jxta-Sim

JXTA-Sim is a simulator that simulates the behaviour of the JXTA lookup algorithm described in (Sandra Garcia Esparza, 2009) over the Dissertation: "*JXTA-Sim A simulator for evaluating the JXTA Lookup Algorithm*". Has is described "*JXTA-Sim is built on top of PlanetSim, a P2P simulator framework, and therefore its design and architecture depend on PlanetSim's design"*. The proposed simulator satisfies the three initial requisites, Scalability, Extensibility and Usability.

Scalability is one of the most important characteristics in a *P2P* algorithm, and JXTA-Sim allow to study the JXTA lookup algorithm performance in scenarios with a large number of nodes, since the development has built on top of a predefined PlanetSim approach that can simulate up to 100.000 nodes and the JXTA-Sim have allowed to test 10.000 nodes. Extensibility provides the simulator the future addiction of the JXTA features. This simulator provides easy usability, and allows the researcher to configure all the parameters necessary to perform the tests. Also the provided results are easy to use and understand, and also provides results in the form of graphs and statistics of the simulated scenarios, and they also code new tests to obtain other results.

## VI. NS-2 and NAM

NS-2 development was supported by the DARPA project through the VINT project, and is currently a well-established network simulator within the research community. Regarding that its initial propose was to be a real network simulator, it has built in support for TCP, routing and multicast protocols over wired and wireless networks. NS-2 has developed in C++, and is described as an event-based simulator for packet-level simulations. The nodes and the characteristics of the communications links are described in TCL scripts, while the protocols are implemented in the C++ language. Since the original version doesn´t include any visualisation tool, the chosen alternative is to use the network animator "*NAM*" witch provide visualizations, and was developed in TCL/TK. NS-2 does not include any visualisation tools by default. NAM is a network animator developed in TCL/TK to provide packet-level animation for the NS-2 simulator. NAM also allows users to arrange network graphs to aid designing and debugging of network protocols (Hossain, 2009). For NS-2 the developers has applied their efforts to make sure the results are accurate, the developed is responsible for verify it´s results and the developed simulator is very complex.

Despite this, the simulator presents simulations that are very closer to match the real network behaviour, and the simulator provides testing and demo scripts to verify its accuracy, providing this simulation with the exclusive ability to confirm the accuracy of the simulation results, with the real network. This simulator provides the Gnutella *P2P* protocol implementation, and also a basic implementation for the BitTorrent protocol, such as the HTTP protocol used for tracker communication. NS-2 needs to use other software to provide the Java interoperability, multi-tier topologies are not known to be compatible, and the scalability is also unknown, since the only registered simulation has implemented with 600 nodes. Extensive documentation, help files and additional documents can be found over an active community that also provides support via mailing lists.

## VII. OMNeT++

OMNeT++ is a discrete event-based simulator, with a component-based and modular architecture. It has embeddable a simulation kernel and the utility class is written in C++, which is most cases is used for random number generation and statistics collection amongst other functions. It is described has a simulation environment with GUI support, and it´s mainly use is for network simulation, but it can also for queuing networks, and multiprocessors. The components for network simulations are not provided, instead there are simulation models and frameworks, and with are used alongside with OMNeT++ (Hornig, 2008).

OMNeT++ has developed with a large contribution on protocols that include IP, UDP, TCP, 802.11, Ethernet, PPP, IPv6, OSPF, and RIP amongst others. Many of this protocols may be used when simulation a P2P network. The BitTorrent protocol has developed (Konstantinos Katsaros, 2009) .The simulator can scale up to 1000 nodes in the swarming simulation, and multi-tier topologies are supported. The java interoperability is provides by the JSimpleModule, which provides the OMNeT++ modules to be written in Java. With this feature some coding conventions need to be adhered to the code and the developer need to have this into account. There are several publications over the website, or through mailing list, a wiki, and a forum for support.

# VIII. OverSim

OverSim is a flexible overlay network simulator framework, and shares many characteristics with OMNeT++ since its modular architecture was based on the OMNeT++, and is also written in C++. The simulation models that can use the network layers from the MAC layer and above are driven from the use of the INET underlay model, which was derived from the INET framework used by OMNeT++. It is possible to simulate the effects of heterogeneous access networks, regarding the bandwidth, delay, packet loss, and queuing effects. The *P2P* protocols available for this simulator are Chord, Kademila, Pastry, Bamboo, Koorde and Broose overlay network protocols. OverSim provides several common functions for structured *P2P* networks to facilitate the implementation of additional protocols and to make them more comparable. The support is provided through mailing list and documentation for installation, usage and development. The simulator is actively developed and open to contributions. The simulator is set to scale up to 100.000 but and no accuracy for the simulator is known (Ingmar Baumgart, 2009).

# IX. GPS

GPS is described as a message-level simulator, which works on the application-lever of the network stack and is written in Java. It provides an extensible object oriented framework that allows modelling of alternative *P2P* protocols, alternative network models, and alternative flow-level models. It provides accurate P2P simulation and allows the implementation of custom protocols. The BitTorrent protocol is available and it appears to be fully implemented, including choking algorithms and support for file transfers. GPS provides a GUI for visualisation of the simulations it has also added delay parameters for the TCP packets and presents real world similarity tests. The support is provided by the developers and some Javadocs, but the project seems to be inactive. (*http://www.cs.binghamton.edu/~wyang/gps/)*.

# X. AgentJ

This simulator was developed by Dr. Ian Taylor from Cardiff University, as part of the Scalable Robust Self-organizing Sensor (SRSS) network project for the U.S Naval Research Laboratory (NRL). AgentJ provides the ability to simulate real-world Java and extends the NS-2 platform to support the simulation and performance analysis of Java network applications, with minor changes to the source code.

"*It supports transport protocols (e.g. UDP unicast and multicast) and timer utilities, which are commonly used in real-world network applications and simulations*" at (Ian Taylor, 2006). This simulator lack the representation over the complexity of architectures such as *JXTA*, since it is a lightweight *peer-to-peer* infrastructure used to developed *P2P* style applications. No visualisation is provides for the simulation, so a third party software is necessary. All the support is provided by an extensive user manual, working code examples and extensive documentation.

## XI. PeerfactSim.KOM

PeerfactSim.KOM is a discrete event based *P2P* Simulator written in Java. It was launched as a project at the Multimedia Communications Lab (KOM) for the simulation of large-scale *peer-to-peer* systems. This framework-like approach has been designed on the concept of pluggable layers, since all components are replaceable to deal with all the possible requirements of the future *peer-to-peer* systems. PeerfactSim.KOM has an integrated churn-generator which determines the dynamic of the peers based on a given distribution, and the simulation experiments can be visualized or written in an output that can be read by *GnuPlot*. The provided PeerfactSim.KOM Visualization has limits regarding the simulations of network´s greater than 80 nodes, so has alternative it's possible to use Gnuplot.

It provides some implemented overlays, such as Chord TON, Kademlia (full version), CAN, globase.KOM, Omicron, Gnutella. This is classified has a message level simulator, since it deals with packet loss, propagation delay and retransmission, and having the ability to simulate overlay networks, it´s scalability over Gnutella is for $10^6$ and at Kademlia is for $10^5$. With this simulator, it´s possible to set many of the features registered for the JXTA algorithm, and there is the possibility to defining the scenarios and simulation based on an XML file. The site for this simulator provides documentation for the simulator and the visualization, but there is a lack of tutorials or discussion groups (http://peerfact.kom.e-technik.tu-darmstadt.de/de/).

# ANNEX B IMPLEMENTATION DETAILS

In this section is discussed all the considerations and decisions performed, on the implementation of the election mechanism. For testing the algorithm it is first necessary to select a simulator. The next session provides the reasons inherent in the selection of the simulator. After selected the simulator is described the structures and decisions for creating the election mechanism, and the key features associated with the implementation. Further there will be made an overview on the messages created/exchanged between the peers.

## I. The simulator

Simulators are typically used to simulate network communications in particular scenarios or situations, without configuring “real” machines or networks and can help with the development and testing of a network application. Within simulators they are divided into two major groups, the network simulators and the overlay simulators.

Network simulators provide a framework for accurate simulation of network protocols such as *TCP*, *UDP*, *IP*, etc. These simulators model the network at the packet level, considering parameters such as delay, bandwidth, the routing for each packet and other lower-level concerns. Some well-known network simulators are NS-2, PlanetSim, and OMNET++. These simulators perform very well when evaluating network protocols but they do not scale well for a large number of nodes. For instance *Omnet++* can't simulate more than 1000 nodes and *Narses* can´t simulate up to 600. This is due to the overhead added by the network details. On the other hand, overlay simulators are less focused on the lower level and more focused on evaluating the overlay algorithms, at the application level.

Overlay simulators receive this concept since the simulation deal with parameters present at the overlay network. The main characteristic of an overlay network is that is built on top of other existing network, it uses a routing technique to send messages to other peers in that overlay-network, some examples of overlay *peer-to-peer* networks are Skyper, Chatnetworks, Gnutella, BitTorrent. This simulators work at the application level, which means they disregard parts of the TCP/network stack. Several simulators, such as Anthill, PeerSim, P2Psim, Jxta-Sim, OverSim, AgentJ, PeerfactSim.KOM but also GPS that have a mechanism to introduce packet delay, to provide realistic communication characteristics, while others do not.

Typically, network simulators take longer to complete a simulation than overlay simulators. This is because of the calculations made for each packet in the simulated network. Network simulators normally allow a developer to produce a network topology and define delay, bandwidth and connection/traffic characteristics for the peers and links. Network simulators such as *NS-2* have been used for testing *P2P* protocols, while other network simulators, like *OMNeT++* have been forced to produce a simulator specifically designed for *P2P* systems, has the *OverSim* simulator.

*P2P* system development has many common issues found with programming other distributed systems, such as difficulty when debugging. The simulator for the *P2P* network needs to have the ability to implement the election mechanism in Java, and be able to support overlay network since the goal is to evaluate the Super Peer Election. So the simulator is required to:

- Provide support for overlay network.
- Provide visualisations for the overview of the network, and the ability to gather statistics.
- Provide reasonably accurate results in terms of “real-world” performance.
- Have good support and/or documentation.
- Have the ability to interface with Java.

The group of simulators can be divided into two sub-categories: network and overlay simulators. Network simulators provide packet-level simulation of network protocols (TCP, UDP, IP, etc.), even awareness of delays, bandwidth and effects of TCP flows, over realistic Internet topologies. Nevertheless, there is an inherent cost on accounting all these low-level concerns, leading to an requirement over the scalability for big networks. Instead, overlay simulators are usually more interested in evaluating overlay algorithms and its routing behaviour without even taking into account the underlying network layer. The excessive overhead and complexity of network simulators thus imposes an unnecessary burden to overlay evaluators and researchers. Other criteria include the scalability for simulations with a large number of peers. Results or simulator output is also examined at in terms of suitability and usability.

# II. Overview

After the description of the algorithm in Chapter 4, its implementation needs to be idealized and proved to achieve an election that have the specific criterions described in Chapter 2 Chapter 3, the: minimize the use of exchanged messages, local knowledge, distributed decision making, external conditional decision. At the beginning of the implementation, is defined the number of nodes that will be present in the network, the time each nodes joins and/or leave the network and the connections between then. This is defined in a file (*Peers Setup File*) that is used to define the peers and its type (Super or Edge), and other file (*Connections File*) to establish the connection between the peer, this two files established the network topology.

```
at 2 Super JOIN 1 1 1
at 3 Super JOIN 2 1 1
at 4 Super JOIN 3 1 1
at 5 Super JOIN 4 1 1
at 6 Super JOIN 5 1 1
at 7 Super JOIN 6 1 1
at 8 Super JOIN 7 1 1
```

Figure 6-1 - Peers setup file

It's also possible to define if a new peer, that joins the network, will have the role of super or edge, and modify this parameter later. This option is set by a file that the simulator uses to setup the initial peers. Here is as example, in Figure 6-1.

Examining the first line in the setup file gives the simulator information about the time that the peer is set to join the network ("at 2"). The second parameter is the type of peer that will join, in this case all the peers will start has super peers ("*Super*"). This simulator also provides the possibility to the nodes join, leave or fail, so this parameter is set for the nodes at time 2, to join the network ("JOIN"). The last parameter is the node ID[8], meaning the identifier of the peer in the network, in this case "1". The connections setup file sets the number of connections between the peers. An example is show in Figure 6-2. In this setup file, at the first column are all the peers present on the network, for Figure 6-2, the network peers are 1, 2, 3, 4, 5, 6 and 7, and

[8] IDentifier.

after each peer are all the connections established to him. Therefore, for the first line, the peer 1 will have established connections from and to the peers 2 and 3. The same it's visible for the peers 2 and 3. Peer 2 has connections to 2, 3 and 6, and the peer 3 has connections to 1, 2 and 4.

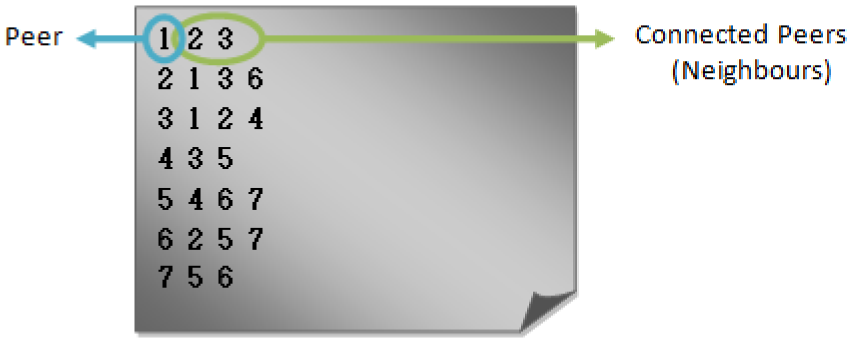


Figure 6-2 - The Connections setup file

This kind of setup offers good control for the construction and maintenance of the network topology, with needs to be controllable. The first step in implementing the algorithm is for each peer to send information about their neighbours to all its neighbours in order to create the view. As a way to create/maintain the information regarding the established links, the algorithm will use the received messages.

Through this, the knowledge of the neighbours is transformed in local knowledge on each node. Information relating to the local knowledge, that is the neighbours on the network, is called the *View*. Each view of a node is unique across the network. Thus while new peer are entering the network, information about the neighbours surrounding each peer is always updated. This is done when a new peer join the network and sends a *JOIN* message to another peer nearby, if the received peer responds, the connections is established, if the recipient does not respond, the new peer sends a new *JOIN* message to connect to other peer nearby until it receives a response. As a peer connects to another, the table that records all the peer connections is updated. This table is called: `PeerView my_RoutingTable`. This information will be sending randomly at each peer, to avoid any future possibility of saturation the network. This occurs in each peer by the function `sendMsgUtilityToAll(){}` independent of the peer state be super or edge. The `PeerView my_RoutingTable` store the peer *`ID`*, and the others peer connected to it. The `PeerView` structure is visible is Figure 6-3.

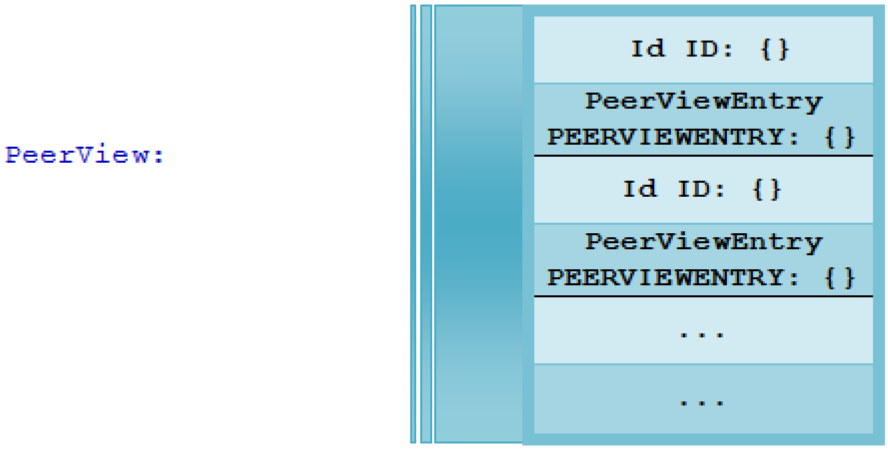


Figure 6-3 - The PeerView content

The peer `ID` and its neighbours are sent over the network using the message: *MY_UTILITY,* to the others neighbours peer. This message has a *TTL* of 2, so it will only be propagated by two peers in the network. When another peer receives the *MY_UTILITY* message it will be propagated to all neighbours, and forwarded to other peers. With the use of *TTL* = 2, the information reaches all peers that are at distance of 2 hops from the peer issuing the message. Thus the local information is sent to a predetermined distance, without the risk of flooding the network.

With this process the neighbours node receives the *MY_UTILITY* message, sets the *TTL* = 1, and resend the message. Once the message is forwarded, each peer also created a dedicated structure to store all messages that are forwarded or received with *TTL* = 1, its name is `Vector<My_Data> vec_Data`, and the process occurs in the function: `storage_msg(msg)`.This vector is responsible to store the source peer `ID`, the `PeerView` and the `n_hops` (Number of hops). With the messages storage, the peer now has information on the other peers that are at distance of two hops, but also its others neighbours. These data will be needed to calculate the future benefits that a combination has, in relation to others. Once this process is concluded, the peer now has a range ser of information to be used in the election mechanism. The next step is to use the stored messages and calculate all the possible combinations. Since the information stored has the number of hops that the message contains when it arrives at the peer. To calculate the combination, all the stored messages will be used, in this way the election can use peers that are at two hops distance and combine with peers that are at one hop distance. The function responsible for this is the `private void combinations` (). A particularity of this mechanism is due to the fact that the combination of none super peer be elected is also considered.

This is because it can happen that for some specific cases the network has better performance for the election without any super peer. All the calculated combination will be stored at the vector: `Vector<My_Combination> vec_Int`.

Using the peer neighbours to calculate all the possible combination, it a relatively low cpu power task. Having a peer with 4 reachable peers at 1 hop and 2 additional peers at 2 hops distance will give 16 possible combinations, since none peer combination is also taken into account. In another case, having 4 peers at 1 hop, and others 4 peers for 2 hops distance provides 256 possible combinations. At a more compact network, for a peer that has 7 peers at 1 hop and additional 12 peers at 2 hops give the total number of combinations of 4096 possible combinations. For each combination a peer cannot be repeated, since it will never appear two times in the network, and the order does not matter. Having a combination with the `ID´s` " 1,5,7 " is exactly the same that having " 1,7,5 " or " 7,5,1 " or even " 5,1,7 ". Each peer combination set, will be constituted by one or several peer/s from the vector: `Vector<My_Data>` *`vec_Data`*. After being created these combinations, it becomes necessary to determine what is/are the best combination/s.

As each combination is unique, it first must be determined which combination offers the most advantage of the indexing mechanism. This is which combination offers the greatest knowledge of the local nodes and has high expansibility. For each case the most advantageous is when the set of nodes that constitute it have the largest number of connections with other nodes. This is considered the first selective process used to select the super peer combination in a distributed network.

This first selection of combination/s will be calculated at two levels. The first level concerns the calculation for their established connections, but only for peers that are at one hop away from the peer in question. In short, it will be calculated the number of peer achievable for the combination, but at a radius of only one hop. So for a combination which only one peer is part and which is at distance of 2 hops, the peer in question will calculate, what is the maximum number of peers that it may have knowledge about? It would not be at all feasible to accomplish this calculation with information about all network nodes, but performing the calculation only for the first hop will also be insufficient, since it does not provide sufficient local knowledge for an efficient method. Once created the vector with all possible combinations, this information will be recorded alongside the number of nodes reachable for the situations of one hop and two hops.

## a. Messages

There were created two types of messages for support of exchanging data between the peers: *MY_UTILITY* and *MY_ELECTION*. The *MY_UTILITY* message exchange information relative to the neighbours of each node. The *MY_ELECTION* message is used only for the neighbourhood information on the outcome of the election. Sending this message will be conducted for all peers that are present in the vicinity so that the outcome of the election arrives to the peer/s to be elected. It could have been a selective choice of destination in order to avoid sending the message to the others peers, such optimization does not represent an advantage to the algorithm, since this message will not take on too much bandwidth between peers, or consume more battery power on the device. The messages used as previously explained are only of two types: *MY_UTILITY* and *MY_ELECTION.*

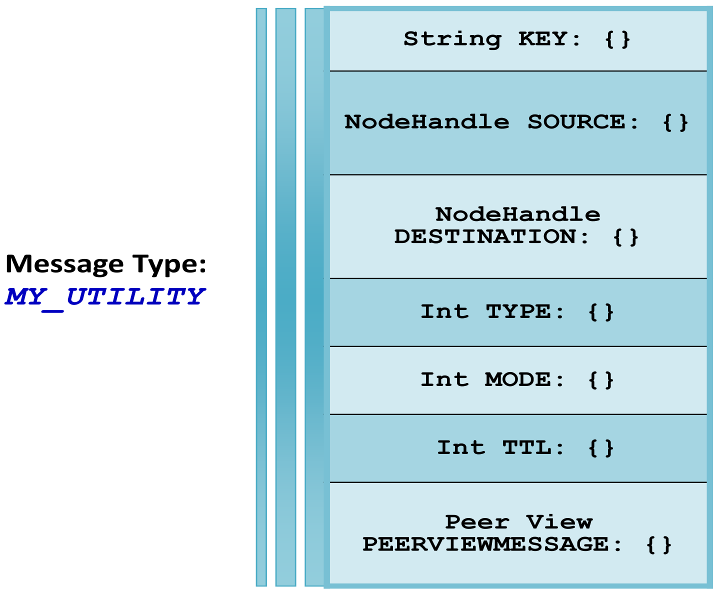


Figure 6-4 - Message *MY_UTILITY* used to exchange information about the node and its neighbours

These two types of messages have on their content all the information needed for the election to occur. The messages exchanged consist of essential fields for their propagation in the network as the Source, Destination and Type. In case the MY_UTILITY message has additionally the fields: Key, Mode, PeerView and Hops.

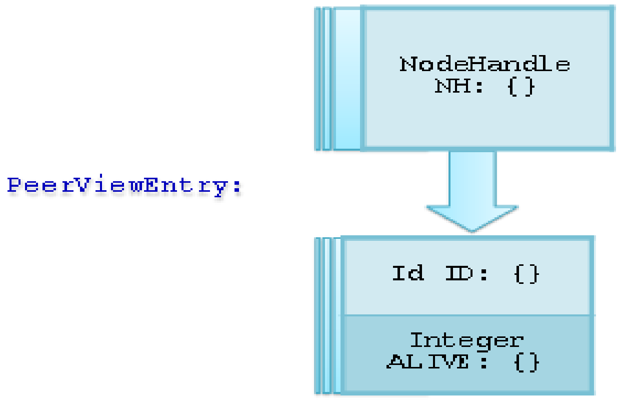


Figure 6-5 - PeerViewEntry content

Has the messages are received by the node in real-time, information about the nodes that are on the network will need to be storage. For this the node will have a local vector called `vec_Data`, used to store all this information about the surrounding nodes. This information it's extremely valuable, since it contains the source for calculating the iterations.

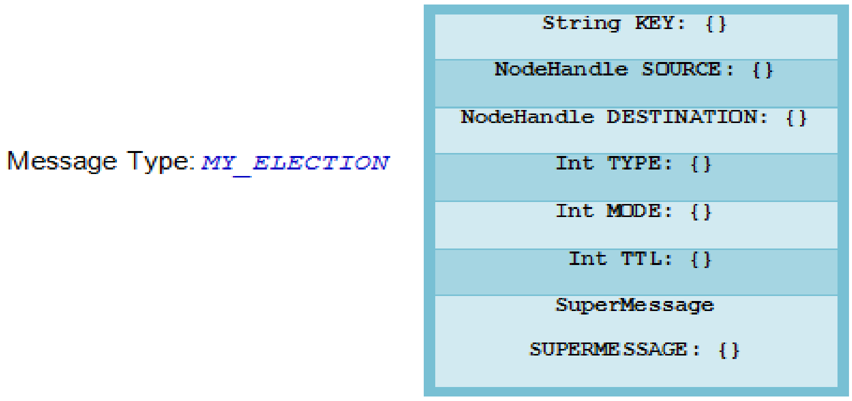


Figure 6-6 - Message *MY_ELECTION* used to inform the neighbours about the Super Peer

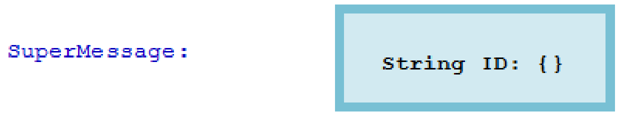


Figure 6-7 - Content of the *MY_ELECTION* message

One issue is how often the peers exchange this information. Obviously, higher frequency means higher accuracy and more traffic overhead. For this design, we employ an event-driven policy in which information exchange is invoked whenever a peer receives a new message.

# Bibliography

**Ahmed M. Mahdy, Jitender S. Deogun, and Jun Wang. 2007.** A Dynamic Approach for the Selection of Super Peers in Ad Hoc Networks. Texas A&M University-Corpus Christi, University of Nebraska-Lincoln : Proceedings of the Sixth International Conference on Networking (ICN'07), 2007.

**Alberti, Antônio Marcos. 2009.** Convergência Digital em Telecomunicações: Das Redes Especializadas à Internet do Futuro. 2009.

**Alberto Montresor, Márk Jelasity. 2009.** PeerSim: A Scalable P2P Simulator. University of Trento, Italy and University of Szeged and HAS, Hungary : http://peersim.sf.net, 2009.

**Alcatel-Lucent, Dimitri Papadimitriou. 2009.** Future Internet: The Cross-ETP Vision Document. 8 1, 2009.

**Alliance, Open Handset. 2008.** Android 2.0 Platform Highlights | Android Developers. *Android Developers.* [Online] Google Inc, 16 April, 2008. [Cited: 21 October, 2010.] http://android-developers.blogspot.com/2010/05/android-22-and-developers-goodies.html.

**Alper Tugay Mızrak, Yuchung Cheng, Vineet Kumar and Stefan Savage. 2003.** Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup. University of California, San Diego : Proceeding WIAPP '03 Proceedings of the The Third IEEE Workshop on Internet Applications , 2003.

**Anis Ismail, Mohamed, Nachouki, Gilles and Hajjar, Mohammad. 2009.** Efficient Super-Peer-Based Queries Routing. Lyon, France : Proceeding MEDES '09 Proceedings of the International Conference on Management of Emergent Digital EcoSystems , 2009. pp. 91-98.

**Anurag Singla, Christopher Rohrs, Lime Wire LLC. 2002.** *Ultrapeers Another Step Towards Gnutella Scalability.* 2002.

**Associates, PricewaterhouseCoopers and Wilkofsky Gruen. 2009.** World Digital Media Trends 2009. 2009.

**Azzedine Boukerche, Regina Araujo and Anis Zarrad. 2009.** A Dynamic Ultrapeers Selection Policy for Collaborative Virtual Environments over Mobile Ad Hoc Networks. 2009.

**B. Traversat, A. Arora, M. Abdelaziz, M. Duigou, C. Haywood, J.-C. Hugly, E. Pouyoul, and B. Yeager. 2003.** Project JXTA. [ed.] Sun Microsystems. *Project JXTA 2.0 super-peer virtual network.* Citeseer : Sun Microsystems, Inc., May, 2003. Technical report.

**Bartosz Biskupski, Jan Sacha, Dominik Dahlem, Raymond Cunningham, René Meier, Jim Dowling, Mads Haahr. 2009.** Decentralising a service-oriented architecture. *Peer-to-Peer Networking and Applications.* The Netherlands, Ireland and Sweden : Springer, 2009. Vol. 3.

**Ben Y. Zhao, Yitao Duan, Ling Huang, Anthony D. Joseph, and John D. Kubiatowicz. 2002.** Brocade: Landmark Routing on Overlay Networks. 2002.

**Bernard Traversat, Mohamed Abdelaziz, Dave Doolin, Mike Duigou, Jean-Christophe Hugly, Eric Pouyoul. 2002.** Project JXTA-C: Enabling a Web of Things. *Project JXTA.* USA : Sun Microsystems, Inc., 2002.

**Beverly Yang Hector, Garcia-Molina. 2002.** Designing a Super-Peer Network. Stanford University : Computer Science Department, 2002.

**Bin Tang, Zongheng Zhou, Anand Kashyap and Tzi-cker Chiueh. 2005.** An integrated approach for P2P file sharing on multi-hop wireless networks. 2005.

**C. Mastroianni, D. Talia, and O. Verta. 2005.** A super-peer model for building resource discovery services in grids: Design and simulation analysis. *In Proceedings of the European Grid Conference,.* Amsterdam, The Netherlands : Springer-Verlag, February, 2005. Vol. 3470 of Lecture Notes in Computer Science, pp. 132-143.

**December 2003.** Challenges in the migration to 4G mobile systems. City University of Hong Kong : IEEE Communications Magazine, December 2003. pp. 54-59.

**Changyong Niu, Jian Wang and Ruimin Shen. 2005.** A Topology Adaptation Protocol for Structured Superpeer Overlay Construction. Shanghai, China : Springer Berlin / Heidelberg, 2005. Vol. 3795/2005, Grid and Cooperative Computing - GCC 2005.

**Chevalier, MB. 2005.** 3-D real-virtual worlds for health and healthcare. University of Plymouth : http://www.slideshare.net, 2005.

**Csaba Simon, Róbert Szabó, Péter Kersch, Balázs Kovács, Alex Galis, Lawrence Cheng. 2005.** Peer-to-peer management in Ambient Networks. 2005.

**Cuibo Yu, Xuerong Gou, Chunhong Zhani, Yang Ji. 2009.** Study on Supenode Election Algorithm in P2P Network Based upon District Partitioning. Beijing, P.R. China : Proceeding ICCSN '09 Proceedings of the 2009 International Conference on Communication Software and Networks , 2009.

**Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins , Zhichen Xu. 2003.** Peer-to-Peer Computing. HP Laboratories Palo Alto : Hewlett-Packard Company, 3 July, 2003.

**Dowling, Jan Sacha and Jim. 2005.** A Gradient Topology for Master-Slave Replication in Peer-to-Peer Environments. Trinity College Dublin, Ireland : Springer-Verlag Berlin Heidelberg, 2005.

**Ehrich, Hans-Dieter. 2009.** Peer-to-Peer Overlay Data Management Part III. Technische Universität Braunschweig in Germany : Institute for Information Systems, 2009. Vols. http://www.ifis.cs.tu-bs.de.

**Ekaterina Chtcherbina, Marquart Franz. 2002.** Peer-to-Peer Coordination Framework (P2PC): Enabler of Mobile Ad-Hoc Networking for Medicine, Business, and Entertainment. Austria : Proceedings of International Conference on Advances in Infrastructure for Electronic Business, 2002.

**Emna Salhi, Mohamed Karim Sbai and Chadi Barakat. 2009.** Neighborhood selection in mobile P2P networks. France, Tunisie : Dans Algotel, 2009.

**Eng Keong Lua, Xiaoming Zhou, Jon Crowcroft, Piet Van Mieghem. 2007.** Scalable multicasting with network-aware geometric overlay. Butterworth-Heinemann Newton, MA, USA : Elsevier B.V., 2007.

**Erkki Harjula, Jussi Ala-Kurikka, Douglas Howie, Mika Ylianttila. 2006.** Analysis of Peer-to-Peer SIP in a Distributed Mobile Middleware System. [ed.] MediaTeam Oulu Group. University of Oulu, Finland : IEEE, 2006. Department of Electrical and Information Engineering.

**Fernanda P. Franciscani, Marisa A. Vasconcelos, Rainer P. Couto, Antonio A.F. Loureiro. 2004.** (Re)configuration algorithms for peer-to-peer over adhoc networks. Minas Gerais : Journal of Parallel and Distributed Computing, 30 November, 2004.

**Fonseca, Hubert. 2008.** COMPARTILHAMENTO DE CONTEÚDO EM REDES. 2008.

**G. P. Jesi, A. Montresor, and Ö. Babaoglu. 2006.** Proximity-aware superpeer overlay topologies. *In Proceedings of the 2nd IEEE International Workshop on Self-Managed Networks, Systems, and Services.* IEEE Transactions on Network and Service Management : Springer, June, 2006. Vol. 3996 of Lecture Notes in Computer Science, pp. 43-57.

**Gabriel André Duquesnois Dubois Brito, Ana Maria de Carvalho Moura. 2005.** ROSA - P2P: a Peer-to-Peer System for Learning Objects Integration on the Web. Engineering System Department - Rio de Janeiro : Proceeding WebMedia '05 Proceedings of the 11th Brazilian Symposium on Multimedia and the web, 2005.

**Georgios Pitsilis, Panayiotis Periorellis, and Lindsay Marshall. 2004.** A Policy for Electing Super-Nodes in Unstructured P2P Networks. University of Newcastle upon Tyne, U.K. : Springer-Verlag Berlin Heidelberg, 2004.

**Georgios Tselentis, John Domingue, Alex Galis, Anastasius Gavras, David Hausheer, Srdjan Krco, Volkmar Lotz, Theodore Zahariadis. 2009.** *TOWARDS THE FUTURE INTERNET.* Amsterdam : IOS Press, 2009. 978-1-60750-007-0.

**Gérald Santucci, Sebastian Lange. 05 September, 2008.** Internet of Things in 2020, A ROADMAP FOR THE FUTURE. *Internet of Things 2020.* 05 September, 2008.

**Gisik Kwon, Kyung D. Ryu. 2003.** An Efficient Peer-to-Peer File Sharing Exploiting Hierarchy and Asymmetry. Arizona State University : Proceeding SAINT '03 Proceedings of the 2003 Symposium on Applications and the Internet, 2003.

**Harvey, Mike. 2008.** The future of social networking: mobile phones. *The Times.* [Online] 9 May, 2008. [Cited: 21 09, 2010.] http://technology.timesonline.co.uk/tol/news/tech_and_web/article3897340.ece.

**Heng Tao Shen, Yan Feng Shu, and Bei Yu. 2004.** Efficient Semantic-based Content Search in P2P Network. Department of Computer Science, National University of Singapore : IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2004.

**Heslep, David B. 2006.** DIGITAL MEDIA ANALYSIS OF GNUTELLA PEER-TO-PEER NETWORKS: LIMEWIRE CASE STUDY. MARYLAND, COMPUTER FORENSICS LABORATORY : MARYLAND STATE POLICE, 2006.

**Hornig, András Varga and Rudolf. 2008.** AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT. Budapest, Hungary : Proceeding Simutools '08 Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, 2008.

**Hossain, Teerawat Issariyakul and Ekram. 2009.** Introduction to Network Simulator NS2. Canada : Springer, 2009. 978-0-387-71759-3.

**Hu, Zhiyong Xu and Yiming. 2003.** SBARC: A Supernode Based Peer-to-Peer File Sharing System. University of Cincinnati, OH : IEEE Computer Society, 2003. 0-7695-1961-X.

**Ian Taylor, Brian Adamson, Ian Downard, Joe Macker. 2006.** AgentJ: Enabling Java for NS-2 Simulations of Distributed Network Applications. Louisiana State University, Washington, DC : IEEE, 2006.

**Ingmar Baumgart, Bernhard Heep, Stephan Krause. 2009.** OverSim: A scalable and flexible overlay framework for simulation and real network applications. Institute of Telematics, Universit¨at Karlsruhe : Proceedings of the 9th International Conference on Peer-to-Peer Computing (IEEE P2P'09), 2009.

**Innovations, Logical. 2007.** Logical Innovations. *http://logicalinnovations.com/about_us.html.* [Online] 2007. [Cited: 6 3, 2010.] http://logicalinnovations.com/p2p_tech.html.

**International Engineering Consortium. 2007.** WPF: Tree and Tabular Combined Notation (TTCN). *IEC: On-Line Education.* [Online] 2007. [Cited: 5 Outubro, 2008.] http://www.iec.org/online/tutorials/ttcn/.

**J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. 2000.** Oceanstore: An architecture for globalscale persistent storage. *In Proceedings of the 9th international Conference on Architectural Support for Programming Languages and Operating Systems.* 2000. pp. 190-201.

**J. Liang, R. Kumar, and K. Ross. April 2005.** The KaZaA overlay: A measurement study. Elsevier : Computer Networks Journal, April 2005. 50:842-858.

**Jan Sacha, Jim Dowling, Raymond Cunningham, and René Meier. 2006.** Discovery of Stable Peers in a Self-organising Peer-to-Peer Gradient Topology. Distributed Systems Group, Trinity College, Dublin : IFIP International Federation for Information Processing, 2006.

**Jim Dowling, Jan Sacha, Raymond Cunningham and Ren´e Meier. 2006.** Using Aggregation for Adaptive Super-Peer Discovery on the Gradient Topology. Distributed Systems Group, Trinity College, Dublin : Springer Berlin / Heidelberg, 2006.

**Jordi Pujol Ahulló, Pedro García López, Marc Sànchez Artigas, Marcel Arrufat Arias, Gerard París Aixalà, Max Bruchmann. 2007.** PlanetSim: An extensible framework for overlay network and services simulations. Tarragona, Spain and Darmstadt, Germany : Proceeding SAC '09 Proceedings of the 2009 ACM symposium on Applied Computing , 2007.

**Konstantinos Katsaros, Vasileios P. Kemerlis, Charilaos Stais and George Xylomenos. 2009.** BitTorrent Module for the OMNeT++ Simulator. Mobile Multimedia Laboratory, Department of Informatics at Athens University of Economics and Business, Athens, Greece : PROCEEDINGS OF THE IEEE MASCOTS, 2009.

**Levene, Mark. 2010.** An Introduction to Search Engines and Web Navigation. New Jersey : John Wiley and Sons, 2010. 047052684X.

**Li Xiao, Zhenyun Zhuang, and Yunhao Liu. 2005.** Dynamic Layer Management in Superpeer Architectures. 2005. Vol. 16, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS.

**Ling-Jyh Chen, Shanky Das, Mario Gerla, Alok Nandan. 2006.** Moving between infrastructure and ad hoc wireless networks: 'opportunistic' mobile middleware. UCLA : Computer Science Department, 2006.

**Lua, Eng Keong and Zhou, Xiaoming. 2007.** Network-aware SuperPeers-Peers Geometric Overlay Network. Hawaii, USA : IEEE, 2007.

**M. Kleis, E. K. Lua, and X. Zhou. 2005.** Hierarchical peer-to-peer networks using lightweight superpeer topologies. *In Proceedings of the 10th IEEE Symposium on Computers and Communications.* Proceeding ISCC '05 Proceedings of the 10th IEEE Symposium on Computers and Communications : IEEE Computer Society, 2005. pp. 143-148.

**M. Siddiqui, A. Villazón, J. Hofer, and T. Fahringer. 2005.** Glare: A grid activity registration, deployment and provisioning framework. In Proceedings of the ACM/IEEE SC2005 Conference on High Performance Networking and Computing : IEEE Computer Society, 2005. pp. 52-67.

**Mankins, John C. 1995.** Technology Readiness Levels. White Paper on Technology Readiness Levels : Advanced Concepts Office - Office of Space Access and Technology - NASA, 6 Abril, 1995.

**Marc Sánchez-Artigas, Pedro García López and Skarmeta, Antonio F. Gómez. 2008.** *On the Feasibility of Dynamic Superpeer Ratio Maintenance.* Spain : IEEE, 2008.

**Marcel C. Castro, Andreas Kassler, Gabriel Kliot, Roy Friedman, Rapha¨el Kummer, Peter Kropf, Pascal Felber. 2009.** Minimizing DHT Routing Stretch in MANETs. 2009. Extended Abstract.

**Mehta, Vadan. 2009.** 5G Wireless Architecture. 2009. p. 8.

**Meier, S. Garcia-Esparza and R. 2010.** JXTA-Sim: Simulating the JXTA Search Algorithm. *in The IEEE International Symposium on Distributed Computing and Artificial Intelligence 2010 (DCAI'10): Springer Verlag - Advances in Intelligent and Soft Computing.* 2010.

**Mert Akdere, Cemal Cagatay Bilgin, Ozan Gerdaneri, Ibrahim Korpeoglu, Ozgur Ulusoy, Ugur Cetintemel. 2006.** A comparison of epidemic algorithms in wireless sensor networks. Journal Computer Communications : Elsevier B.V., 2006. Vol. 29.

**Montresor, Alberto. 2004.** A Robust Protocol for Building Superpeer Overlay Topologies. Department of Computer Science, University of Bologna, Italy : IEEE Computer Society, 2004.

**Nao Chen, Ruimin Hu, Yongqiong Zhu, Zhaopin Wang. 2010.** Constructing Fixed-Ratio Superpeer-based Overlay. National Engineering Research Center for Multimedia Software, Wuhan University, Wuhan, China : IEEE, 2010. Vol. 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT).

**Onofre, Sérgio Miguel da Silva. 2007.** *Plataforma para testes de conformidade de sistemas baseados em módulos conceptuais STEP.* Departamento de Engenharia Electrotécnica, Universidade Nova de Lisboa - Faculdade de Ciências e Tecnologia. 2007. Dissertação de Mestrado.

**Ozalp Babaoglu, Hein Meling, Alberto Montresor. 2001.** Anthill: a Framework for the Development of Agent-Based Peer-to-Peer Systems. Department of Computer Science, University of Bologna : IEEE Computer Society, 2001.

**Papapetrou, Odysseas. 2008.** Full-text indexing and Information Retrieval in P2P Systems. Research Center, Leibniz Universität Hannover, Germany : Proceeding Ph.D. '08 Proceedings of the 2008 EDBT Ph.D. workshop, 2008.

**Ping Ge, Hailong Cai. 2008.** Providing Differentiated QoS for Peer-to-Peer File Sharing Systems. University of Central Florida, Google : Newsletter ACM SIGOPS Operating Systems Review, 2008.

**Q. Lv, S. Ratnasamy, and S. Shenker. 2002.** Can heterogeneity make gnutella scalable? *In Proceedings of the 1st International Workshop on Peer-to-Peer Systems.* 2002. Vol. 2429, Lecture Notes in Computer Science, pp. 94-103.

**R. Szabo et al. 2005.** Dynamic Network Composition for Ambient Networks. Germany : EURESCOM Summit 2005 - Ubiquitous Services and Applications, 2005. Accepted for publication at EURESCOM 2005.

**Rashmi Ranjan Rout, K. Shiva Rama Krishna, Krishna Kant. 2008.** A Centralized Server Based Cluster Integrated Protocol in Hybrid P2P Systems. Allahabad, India : Proceeding ICETET '08 Proceedings of the 2008 First International Conference on Emerging Trends in Engineering and Technology, 2008.

*Resource Location in P2P Systems.* **Alveirinho, João Pedro Fernandes. 2009.** 2009, Networks, p. 31.

**S. Corson, J. Macker. 1999.** Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. Naval Research Laboratory, University of Maryland : RFC Editor, 1999.

**S. Guha, N. Daswani, and R. Jain. 2006.** An Experimental Study of the Skype Peer-to-Peer VoIP. *In Proceedings of the 5th International Workshop on Peer-to-Peer Systems.* 2006.

**S. Johnstone, P. Sage, P. Milligan. 2005.** iXChange – A Self-Organising Super Peer Network Model. School of Computer Science, Queen's University Belfast, : Proceeding ISCC '05 Proceedings of the 10th IEEE Symposium on Computers and Communications , 2005.

**S. Saroiu, P. K. Gummadi, and S. D. Gribble. July 2003.** Measuring and analyzing the characteristics of napster and gnutella hosts. Journal Multimedia Systems : Springer-Verlag New York, July 2003. Vols. 9(1):170-184.

**Sandra Garcia Esparza, B.Sc. 2009.** JXTA-Sim A simulator for evaluating the JXTA Lookup Algorithm. University of Dublin : Trinity College, 2009.

**Sauve, Eric. 2007.** Communities of Practice: Addressing Workforce Trends Through New Learning Models. *eLearn Magazine.* [Online] 2007. [Cited: 1 9, 2010.] http://www.elearnmag.org/subpage.cfm?article=37-1&section=best_practices.

**Schafersman, Steven D. 1994.** Scientific Thinking and the Scientific Method. *An Introduction to Science.* [Online] Janeiro, 1994. [Cited: 10 Setembro, 2008.] http://www.freeinquiry.com/intro-to-sci.html.

**Schollmeier, Rüdiger. 2001.** A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. München, Germany : Proceeding P2P '01 Proceedings of the First International Conference on Peer-to-Peer Computing, 2001. Institute of Communication Networks.

**Schulzrinne, S. A. Baset and H. April 2006.** An analysis of the Skype peer-to-peer internet telephony Protocol. Proceedings IEEE INFOCOM 2006 25TH IEEE International Conference on Computer Communications (2004) : IEEE, April 2006. Vol. 6, pages 1-11.

**Scott Oaks, Bernard Traversat, Li Gong. 2002.** JXTA in a nutshell. JXTA in a Nutshell : O'Reilly Media, 2002. 978-0-596-00236-7.

**Seet, Boon-Chong. 2004.** Mobile P2Ping: A Super-Peer based Structured P2P System Using a Fleet of City Buses. Singapore-MIT Alliance : Third IEEE International Conference on Pervasive

Computing and Communications Workshops, 2005. PerCom 2005 Workshops. , 2004. 0-7695-2300-5 .

**Subbarao, Madhavi W. 1999.** Ad Hoc Networking Critical Features and Performance Metrics. Wireless Communications Technology Group : Proceeding MSWiM '05 Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, 1999.

**Sun, Jun-Zhao. 2001.** Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing. University of Oulu, Finland : MediaTeam, 2001. p. 6.

**T.I. Wang, K.H. Tsai, and Y.H. Lee. 2004.** Crown: An Efficient and Stable Distributed Resource Lookup Protocol. National Cheng Kung University, Tainan, Taiwan. : Springer-Verlag Berlin Heidelberg, 2004.

**Technology, ISO. Information. 1991.** Open Systems Interconnection, Conformance Testing Methodology and Framework. International Standard IS-9646. ISO. 1991. Vols. CCITT X.290–X.294.

*The Internet of Things.* **Union, International Telecommunications. November 2005.** November 2005.

**Tretmans, Gerrit Jan. 1992.** A Formal Approach to Conformance Testing. *A formal approach to conformance testing.* The Netherlands : Hengelo, 1992. 90–9005643–2.

**Tretmans, Jan. 2001.** An Overview of OSI Conformance Testing. University of Twente : Proceeding MOVEP '00 Proceedings of the 4th Summer School on Modeling and Verification of Parallel Processes, 25 Janeiro, 2001.

**Tsai, Mario Gerla and Jack Tzu-Chieh. 1995.** Multicluster, mobile, multimedia radio network. Los Angeles : Journal Wireless Networks, 1995. Vol. 1 Issue 3.

**Tyson E. Condie, Sepandar D. Kamvar, Hector Garcia-Molina. 2004.** Adaptive Peer-To-Peer Topologies. Stanford University, Stanford : IEEE Press, 2004.

**Virginia Lo, Dayi Zhou, Yuhong Liu, Chris GauthierDickey, and Jun Li. 2005.** Scalable Supernode Selection in Peer-to-Peer Overlay Networks. Network Research Group – University of Oregon :

Proceeding HOT-P2P '05 Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems , 2005.

**Wu, Wei Lou and Jie. 2007.** Toward Broadcast Reliability in Mobile Ad Hoc Networks with Double Coverage. *IEEE TRANSACTIONS ON MOBILE COMPUTING.* FEBRUARY, 2007. Vol. 6, Broadcast, double dominating set, forwarding node, mobile ad hoc networks (MANETs), performance evaluation, pp. 148-163.

**Yifen WEI, Gaogang XIE and Zhongcheng LI. 2007.** A Hierarchical Cross-Layer Protocol for Group Communication in MANET. Penang, Malaysia : Proceedings of the 2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, 2007.

**Yuh-Jzer Jounga, and Zhang-Wen Lin. 2009.** On the self-organization of a hybrid peer-to-peer system. Taiwan : Journal of Network and Computer Applications, 2009. Vol. 33.

**Zahariad. 2008.** The Future Internet: A Content Creation and Media Delivery Perspective. 2008. Draft to be reviewed by a working group in Bled.